



الجمهورية العربية السورية

وزارة التعليم والبحث العلمي

جامعة تشرين - كلية الهندسة المعلوماتية

قسم النظم والشبكات الحاسوبية

مشروع فصلي بعنوان

تحليل ونمذجة الشبكات المعرفة بالبرمجيات

إعداد الطلاب:

حيدر مجيد محفوظ

حسين محمد الفروي

نوح سهيل عبود

صبا حسين علي

صبا أبي سليمان

لين عبود خليل

إشراف:

د.م رضوان دنده

العام الدراسي:

2022/2021

ملخص:

الشبكات المعرفة بالبرمجيات (SDN) هي تقنية متطورة تختلف عن بنية وعمل الشبكات التقليدية. كان هناك إلحاح متزايد بين مطوري الشبكات للتوصل إلى مجموعة جديدة من الأجهزة لدعم الشبكات المعرفة بالبرمجيات ولتنفيذ خدمات أفضل في SDN مقارنة بالشبكات التقليدية. في هذا المشروع قمنا بدراسة البنية التحتية التي تتكون منها الشبكات المعرفة بالبرمجيات و أهم المميزات التي تتصف بها. كما قمنا بالمقارنة بين هذه البنية وبنية الشبكات التقليدية.

في القسم الخاص بالتطبيق العملي قمنا بتحليل ونمذجة شبكات SDN من خلال تطبيق طوبولوجيات SDN باستخدام أداة Mininet و قمنا بإرسال باكيتات عبر الشبكة ودراسة التأخير وسرعة النقل ضمن الشبكة.

والأدوات المستخدمة في المشروع:

تم إجراء التطبيق العملي للشبكة وفق تقنية SDN باستخدام المحاكى Mininet

نحتاج المتطلبات التالية:

- Vmware WorkStation or VirtualBox
- Mininet package with Ubuntu Server
- متحكم POX
- Gnuplot لتقييم الأداء

Abstract:

Software Defined Networking (SDN) is an advanced technology that differs from the architecture and operation of traditional networks. There has been a growing urgency among network developers to come up with a new set of devices to support software defined networks and to implement better services in SDN than in traditional networks. In this project, we study the infrastructure that makes up software-defined networks and the most important features that characterize them. We also compared this structure with that of traditional networks.

In the practical application section, we analyze and model SDNs by applying SDN topologies using Mininet tool. We send packets across the network and study the delay and throughput within the network.

The tools used in the project:

The practical application of the network was carried out according to SDN technology using the Mininet emulator

We need the following requirements:

- Vmware WorkStation or VirtualBox
- Mininet package with Ubuntu Server
- POX Controller
- Gnuplot

الفهرس

1. الفصل الأول: الشبكات المعرفة بالبرمجيات (software-defined networks)	11
1.1 مقدمة	11
2.1 تعريف تقنية ال SDN	12
3.1 البنية التحتية لشبكات ال SDN	13
4.1 المتحكمات controllers	15
5.1 بروتوكول Openfolw	16
6.1 الفوائد التي قدمتها الشبكة المعرفة بالبرمجيات	17
7.1 ال SDN من وجهة نظر أمنية	18
2. الفصل الثاني: الانتقال من الشبكات التقليدية إلى ال SDN وبرتوكول	
Openfolw	19
1.2 مقدمة:	19
2.2 أسباب الانتقال إلى الشبكات المعرفة بالبرمجيات	19
3.2 خطوات ادخال التحكم المبرمج إلى الشبكة	21
1.3.2 المرحلة الأولى: الإدارة الموحدة	21
2.3.2 المرحلة الثانية: استخدام بروتوكول OpenFlow هجين	
	22
3.3.2 المرحلة الثالثة: نشر نموذج SDN الأمثل	22

23.....	4.2 صعوبات الانتقال إلى تقنية ال SDN
24.....	5.2 تعريف بروتوكول OpenFlow
24.....	6.2 بنية شبكة OpenFlow
25.....	7.2 إصدارات openflow
26.....	8.2 OpenFlow enabled switch
27.....	9.2 Flow Table
28.....	1.9.2 اقسام Data Flow Table Entry
28.....	2.9.2 الإجراءات المتخذة بشأن التدفقات
30.....	3.9.2 أنواع التدفقات
31.....	10.2 آلية إعادة توجيه الحزمة في شبكة OpenFlow
32.....	11.2 نلخص عمل البروتوكول OF
33.....	12.2 رسائل الاتصال OpenFlow
33.....	1.12.2 سائل Controller-to-switch
34.....	2.12.2 Asynchronous Messages
34.....	3.12.2 Symmetric Message
35.....	13.2 شرح توضيحي للرسائل المتبادلة في شبكة OpenFlow
36.....	1.13.2 الرسائل بين المبدل ووحدة التحكم
37.....	2.13.2 تبادل الرسائل بين مضيفين
38.....	3. الفصل الثالث
38.....	1.3 المقدم
38.....	2.3 single topology

39.....	Linear topology 3.3
40.....	Tree topology 4.3
41.....	5.3 المتحكمات في الشبكات المعرفة بالبرمجيات
41.....	1.5.3 متحكم pox في SDN
42.....	2.5.3 متحكم NOX
42.....	3.5.3 متحكم RYU
43.....	4.5.3 متحكم BEACON
45.....	5.5.3 متحكم FLOODLIGHT
45.....	6.3 اختبار أداء المتحكمات
46.....	4 الفصل الرابع
.....	1-4 مقدمة حول أداة ال mininet
.....	2-4 السيناريو الأول
.....	1-2-4 المكونات المستخدمة
.....	1-1-2-4 المتحكم
.....	2-1-2-4 المضيف
.....	3-1-2-4 المبدل
.....	4-1-2-4 أداة ال miniedit
.....	5-1-2-4 أداة ال putty
.....	6-1-2-4 xming server
.....	2-2-4 آلية التنفيذ
.....	3-4 السيناريو الثاني
.....	1-3-4 خطوات العمل

فهرس الأشكال

.....	الفصل الأول
14.....	الشكل (1-1)
16	الشكل (2-1)
25.....	الشكل (3-1)
29	والشكل (4-1)
.....	الفصل الثاني
34	الشكل (1-2)
36	الشكل (2-2)
39	الشكل (3-2)
40.....	الشكل (4-2)
41.....	الشكل 5-2
42.....	الشكل (6-2)
44.....	الشكل (7-2)
46.....	الشكل (8-2)
.....	الفصل الثالث
51.....	الشكل (1-3)
53.....	الشكل (2-3)
56.....	الشكل (3-3)
57.....	الشكل (4-3)
57.....	الشكل (5-3)
58.....	الشكل (6-3)
.....	الفصل الرابع

62.....	الشكل (1-4)
63.....	الشكل (2-4)
64.....	الشكل (3-4)
65.....	الشكل (4-4)
66.....	الشكل (5-4)
67	الشكل (6-4)
68.....	الشكل (7-4)
69	الشكل (8-4)
70	الشكل (9-4)

قائمة المصطلحات :

Allied Telesis Management Framework	AMF
Generalized Multiprotocol Lable Switching	GMPLS
Multiprotocol Lable Switching	MPLS
Open networking Foundation	ONF
Software defined network	sdn
Transmission Control Protocol	TCP
Time Length Value	TLV
Virtual local Area Network	VLAN
WIDE AREA NETWORK	WAN

الفصل الأول

الشبكات المعرفة بالبرمجيات (software-defined networks)

1-1 مقدمة:

الشبكات المعرفة بالبرمجيات هي أكثر المواضيع المطروحة للنقاش في السنوات الأخيرة. فقد قَدِّمت العديد من المميزات وساهمت في حل العديد من المشاكل التي عانت منها الشبكات التقليدية. تعتمد الشبكات المعرفة بالبرمجيات على فصل عمليات توجيه البيانات عن عمليات التحكم واتخاذ قرارات التوجيه لجعل التحكم بالشبكة مركزياً وبالتالي تقليل الأخطاء التي من الممكن أن تتعرض لها الشبكات. يُعتبر بروتوكول OpenFlow هو الأكثر شيوعاً واستخداماً في متحكمات الشبكات المعرفة بالبرمجيات للتواصل والتحكم بالمبدلات. باستخدام هذا البروتوكول، يتعلم المبدل معلومات التوجيه من المتحكم ومن ثم تمرير حزم البيانات بالاعتماد على هذه المعلومات.

فتحت الشبكات المعرفة بالبرمجيات (SDN) فرصاً جديدة لعشاق الشبكات لتجريب ونشر طرق مبتكرة لإدارة الشبكات وللتحكم ديناميكياً في توجيه حزم البيانات في شبكاتهم. ومع ذلك ، فإن ما لم يتغير هو الحاجة إلى فهم طوبولوجيا الشبكة والبقاء على اطلاع حول كيفية أداء عناصر الشبكة عبر الطوبولوجيا. ولا يزال هذا العامل عاملاً رئيسياً في تحقيق أهداف الأداء والأمان ، وهي على الأرجح الأسباب التي تجعل أمن الشبكات ومراقبتها يتسمان بأهمية كبيرة في مجتمع الشبكات. مع زيادة التحكم في الشبكة بطريقة مركزية تأتي مسؤولية أكبر لوحدة تحكم SDN لمراقبة ومنع أي هجمات على الشبكة.



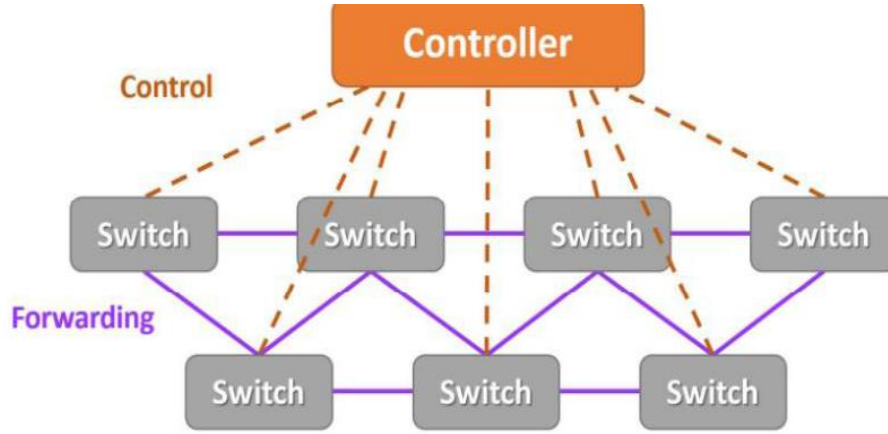
الشكل (1-1) تعريف ال sdn

2-1 تعريف تقنية ال SDN:

اختصار ل (Software Defined Networking) هي التقنية التي يتم فيها فصل ال Data plane عن ال control plane في أجهزة الشبكة، ليصبح دور هذه الأجهزة مقتصر على تمرير البيانات، أما الإدارة والتحكم ستصبح في طبقات جديدة.

في الشبكات التقليدية ، تكون عمليتي تمرير البيانات و تقرير وجهة البيانات (التوجيه) متضمنتان ضمن نفس الجهاز وهذا ما جعلها تبدو معقدة.

في تقنية ال SDN فإن كل الطلبات الخاصة بتوجيه البيانات يتم إنجازها في برمجية خاصة (المتحكم)، أما الأجهزة فهي ببساطة تطبق القرارات التي يرسلها لها المتحكم على رزم البيانات ، ولهذا السبب فإن لا حاجة بأن يكون لأجهزة الشبكة إدراك لمنطق التوجيه ، وإنما فقط تخزين (caching) لقرارات المتحكم بشأن التوجيه.



الشكل (2-1) شبكة ال sdn

3-1 البنية التحتية لشبكات ال SDN:

تقسم إلى ثلاث طبقات موضحة بالشكل هي :

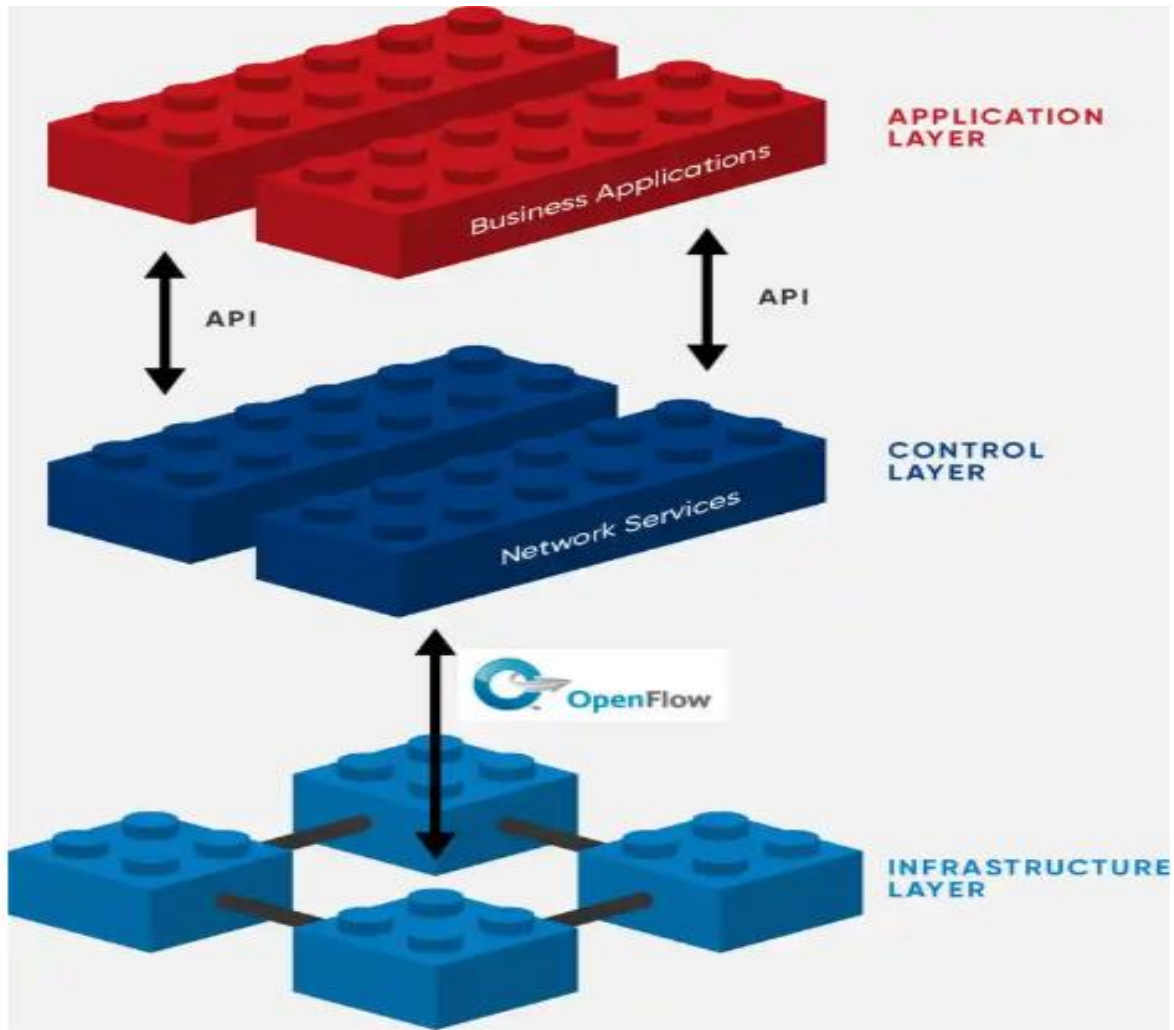
1. طبقة التطبيقات : تتكون من الخدمات والتطبيقات التي تقدمها الشبكة للمستخدم ،وتتواصل هذه الطبقة

مع الطبقة التي تليها (طبقة التحكم) عن طريق واجهات برمجية API

2. طبقة التحكم: تمثل نقطة التحكم المركزية بأجهزة الشبكة . حيث يكون التحكم في الشبكة وإدارتها يتم عن

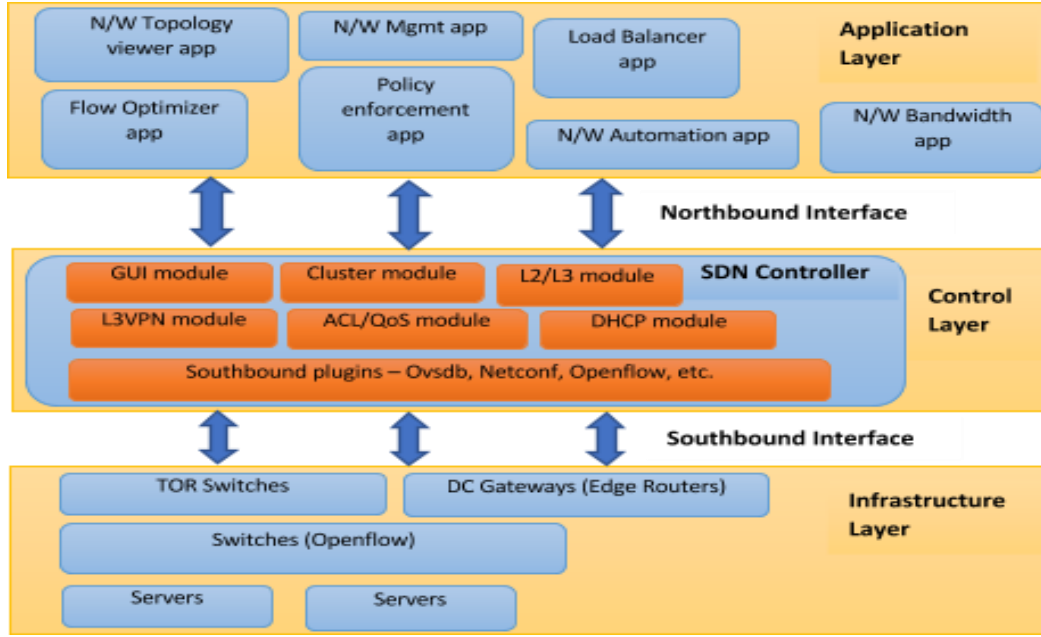
طريق جهاز أساسي يسمى Controller.

3. طبقة البيانات : تمرير البيانات إلى وجهتها.



الشكل (3-1) بنية ال sdn

غالبًا ما يُشار إلى واجهات برمجة تطبيقات بنية SDN بأنها واجهات الشمالية والجنوبية ، تحدد الاتصال بين التطبيقات ووحدات التحكم وأنظمة الشبكات. يتم تعريف الواجهة الشمالية بأنها الاتصال بين وحدة التحكم والتطبيقات ، في حين أن الواجهة الجنوبية هي الاتصال بين وحدة التحكم وأجهزة الشبكة المادية كما هو موضح في الشكل التالي .



الشكل (4-1) نموذج ال sdn تفصيلياً

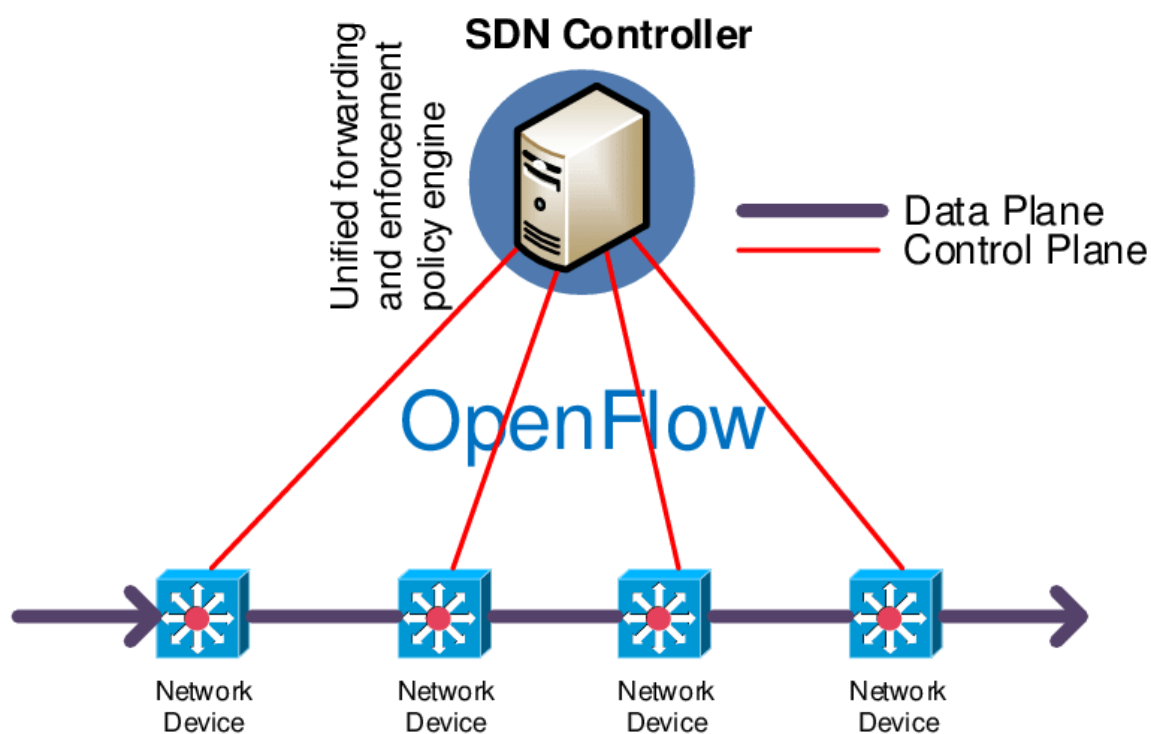
4-1 المتحكمات controllers:

يعتبر المتحكم عقل ال SDN لأننا جمعنا فيه وظيفة control plane المسؤولة عن الخدمات والتطبيقات المطلوبة والتي تم برمجتها من قبل مهندس الشبكات بواسطة API لتنتقل بشكل إعدادات إلى أجهزة الشبكة الأخرى.



الشكل (5-1) متحكمات

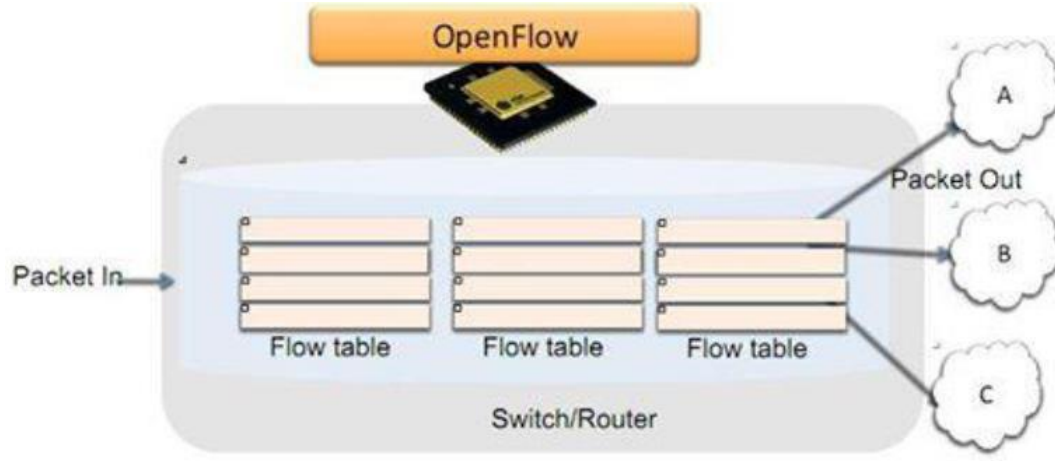
يتواصل جهاز ال controller مع باقي أجهزة الشبكة التي تم نزع وظيفة التحكم منها لينقل لها الخدمات والتطبيقات وذلك باستخدام لغة مشتركة بينهما تسمى البروتوكول openflow



الشكل (1-6) بروتوكول Openfolw

5-1 بروتوكول Openfolw :

هو بروتوكول التواصل بين طبقة التحكم والطبقة التي تليها (طبقة البنية التحتية) فيقوم بإنشاء قناة التواصل بين control plane والذي يمثلها المتحكم controller وال data plane والذي يمثلها switch حيث قاموا بتنظيم جميع المكونات الخاصة بعملية التوجيه ووضعها في إطار أسموه flow tables أخذين بعين الاعتبار جميع المعلومات المشتركة بين مختلف المصنعين vendor كما في الصورة التالية:



الشكل (1-7) بروتوكول Openfolw

تم إنشاء هذا البروتوكول ليكون حلاً للتعامل مع أجهزة السويتش بشكل مفتوح كي يتم استقبال الطلبات من المستوى الأعلى نظراً لكم الهائل من الخدمات التي يتم تحديثها يومياً من قبل المهندسين والباحثين.

1-6 الفوائد التي قدمتها الشبكة المعرفة بالبرمجيات:

- 1- قابلة للبرمجة بشكل مباشر ، و التحكم في الشبكة بشكل مباشر ايضاً.
- 2- يستطيع مهندس الشبكة التحكم في الشبكة بشكل كامل من مكان واحد حيث يقوم بعمل ادارة و تحكم و صيانة من مكان واحد لي انه يوجد شروط في تقنية ال SDN تقوم بتحكم في الشبكة ، حيث يقوم مهندس الشبكة بوضع هذه الشروط.
- 3- تحسين عملية ارسال البيانات في الشبكة حيث من ناحية التوجيه و توزيع الترافيك في الشبكة.
- 4- سهولة صيانة الشبكة و مراقبة الشبكة بشكل اوسع و اسهل ، حيث انه يتواجد وحدة مركزية للتحكم الكامل في الشبكة كلها.
- 5- توفير عدة كبير من أجهزة الشبكة حيث انه نستطيع عمل أجهزة شبكة افتراضية ولكن وهمية وغير موجودة في الواقع.
- 6- سيتواجد شبكات افتراضية برمجية و ستكون اسهل بكثير من أن يكون عدة شبكات موجودة في الواقع الحقيقي ، حيث أنه سيوفر لنا الكثير من الوقت و توفير من ناحية التكلفة و سهولة في الادارة.
- 7- تعمل هذه التقنية مع التقنية التالية مثل . MPLS , GMPLS

8- يستطيع مهندس الشبكة توسيع الشبكة بكل سهولة لي انه بشكل افتراضي و وهمي و هذا يسهل الكثير من العمل على مهندس الشبكة و يكون افضل من أن تكون الشبكة موجودة بشكل حقيقي.

1-7 ال SDN من وجهة نظر أمنية:

ال SDN يوفر وظائف أمان الشبكة لأنه يوفر إعادة توجيه أو تصفية تدفقات حركة المرور استنادًا إلى محتوى الحزمة ، والتي تتطلب حتى الآن أجهزة أمان شبكة إضافية مثل جدران الحماية أو أنظمة كشف التسلل في الشبكات التقليدية. من ناحية أخرى ، نظرًا للفصل المادي التي تقوم به هذه التقنية ، قد توفر SDN موجّهات هجومية إضافية مقارنة بالشبكات التقليدية ، مما قد يؤثر بشدة على توفر الشبكة بشكل عام وكذلك على سرية بيانات الشبكة والتحكم فيها وموثوقيتها وتكاملها.

الفصل الثاني

الانتقال من الشبكات التقليدية إلى الـ SDN وبرتوكول Openflow

2-1 مقدمة:

نقل التحكم الموزع إلى الشبكات التقليدية لا يمكن أن يحدث بشكل فوري أو بين ليلة وضحاها. في الواقع، يمكن أن يحدث الانتقال من شبكة تقليدية إلى شبكة معرفة بالبرمجيات (Software Defined-Network) بطريقة تدريجية. هناك عدد من الأسباب التي تجعل من الصعب للغاية تنفيذ الانتقال المفاجئ إلى شبكة معرفة بالبرمجيات.

كبدائية، فإن مقدار التخطيط المطلوب لضمان استمرار تشغيل جميع خدمات الشبكة الحالية يجعل من الصعوبة الانتقال إليها أي شبكة SDN، هناك تفاصيل لا حصر لها للنظر فيها وتحليلها، والعديد من السلوكيات غير المتوقعة للتنبؤ بها. اختبار الحل الجديد هو أيضا إشكالية حيث لا يمكن اختبار النظام الجديد بأكمله مسبقًا، لأن ذلك يعني إنشاء شبكة بيانات كاملة. من الممكن فقط اختبار مجموعات فرعية محدودة من الشبكة. علاوة على ذلك، تعتبر الخدمات اللوجستية الخاصة باستكمال عملية نقل الشبكة التقليدية إلى شبكة SDN تحدّي صعب.

إن الانتقال إلى شبكة معرفة بالبرمجيات هو ضرورة عملية ذات فائدة كبيرة، ولكن ينبغي أن

تحدث في سلسلة من الخطوات، كل منها يتناول حاجة معينة.

2-2 أسباب الانتقال إلى الشبكات المعرفة بالبرمجيات:

شبكة البيانات هي العصب المهم في أي مؤسسة، والسبب الوحيد لتغيير عمل الشبكة هو لجعلها تخدم المؤسسة بشكل أفضل. تتمثل نقطة البداية في هذا الانتقال في تحديد الطرق التي تتوقع الشركة من خلالها الحصول على قيمة أكبر من شبكتها الجديدة المعرفة بالبرمجيات.

تتمثل الفوائد الرئيسية لإدخال التحكم البرمجي إلى الشبكة في:

1. تقليل التكاليف:

يمكن تخفيض تكاليف تشغيل الشبكة بطريقتين رئيسيتين: تبسيط المهام المعقدة، وأتمتة المهام الروتينية. باستخدام برنامج ذكي كالمتحكم controller هو المفتاح لتحقيق كليهما. يُساعد إدخال التحكم المبرمج في تبسيط المهام المعقدة مثل:

1. استكشاف المشاكل وإصلاحها
2. إعادة تشكيل الشبكة
3. توقع التنافس على الموارد
4. إن توحيد الشبكة في ظل التحكم المبرمج يمكن من أتمتة موثوقة للأنشطة الروتينية، مثل:
5. النسخ الاحتياطية لإعدادات الشبكة
6. ترقية البرمجيات
7. استبدال وحدات معينة

2. زيادة المرونة:

لقد أدى ظهور التنقل Mobility وانتشار التطبيقات إلى زيادة عدد الإمكانيات المطلوبة من الشبكة. حيث يجب أن يتم تخديم هذه التقنيات، وبطريقة تتوافق مع السياسات التنظيمية للشبكة. ونتيجة لذلك، يجب أن تعمل شبكات اليوم بطريقة مرنة بشكل متزايد تأخذ في الاعتبار عدداً كبيراً من العوامل بما في ذلك هوية المستخدم ومحتوى التطبيق وسياسات الاستخدام. يتطلب الحصول على مستويات متناسقة من إدراك المحتوى الشبكي والسلوك الديناميكي عبر الشبكة بأكملها قدرات لا تمتلكها عادةً عقد الشبكة، وهو ما يقع على عاتق المتحكم المركزي ليقوم به.

3. تحسين تجربة المستخدم:

زيادة المرونة والاستجابة والحماية من التهديدات، مع تقليل وقت التوقف عن العمل، كل ذلك يؤدي إلى تحسين تجربة المستخدم. غالبية المستخدمين لا يعطون إلا القليل من التفكير للبنية التحتية للشبكة إلا عند حدوث خلل. ما يريده المستخدم هو وصول عالي الأداء إلى خدماتهم وتطبيقاتهم متى وأينما يحتاجون إليها. أفضل طريقة لتوفير هذه التجربة هي تحسين سلوك الشبكة، عن طريق التحكم الذكي المتكثف.

2-3 خطوات ادخال التحكم المبرمج إلى الشبكة:

إدخال التحكم المبرمج هو عملية من ثلاث مراحل:

1. تطبيق إدارة الموحدة
2. استخدام بروتوكول OpenFlow هجين
3. نشر نموذج SDN الأمثل

2-3-1 المرحلة الأولى: الإدارة الموحدة

تتمثل الخطوة الأولى نحو تحقيق التحكم المبرمج في الشبكة في توحيد إدارة الشبكة، بحيث يمكن إدارتها ككيان واحد. يقترح Allied Telesis إدارة الشبكة بإطار Allied Telesis Management Framework (AMF)، الذي يشتمل على إطار إدارة ذكي في الشبكة. يوفر إطار AMF ما يلي:

1. إنشاء تلقائي لمسارات الاتصال عبر الشبكة
 2. أتمتة المهام الروتينية مثل النسخ الاحتياطية وترقيات البرامج وفحص سلامة الجهاز
 3. استبدال الوحدات التي توقفت عن العمل وتوفير عقد شبكة جديدة
 4. تكوين إعدادات عقد الشبكة المتعددة في وقت واحد وبشكل متزامن
 5. جمع معلومات استكشاف الأخطاء وإصلاحها من العقد المتعددة في نفس الوقت
- تحقق هذه الميزات الجزء الأكبر من احتياجات خفض التكاليف عند الانتقال إلى شبكة يتم التحكم فيها بالبرمجيات. علاوة على ذلك، يسهل إطار AMF تحقيق تطبيقات التحكم. يعد إدخال إطار AMF في شبكة موجودة أمرًا بسيطًا، فهو يتطلب بعض أوامر التكوين على كل جهاز. الأهم من ذلك، يمكن أن يعمل إطار AMF في شبكة ذات عقد من مزودين مختلفين. يمكن توصيل اتصالات AMF عبر شبكات لا تدعم AMF، مما يسمح بتوصيلها عبر الإنترنت أو خدمة WAN أخرى.

2-3-2 المرحلة الثانية: استخدام بروتوكول OpenFlow هجين

في حين يوفر إطار AMF إدارة شبكة موحدة، فإن المفتاح للتوجيه الآلي لدفق بيانات الشبكة هو بروتوكول OpenFlow. بروتوكول OpenFlow هو واجهة مفتوحة المعايير تسمح بالتحكم الدقيق في كيفية قيام جهاز

الشبكة بتوجيه أو رفض دفع البيانات. بمجرد تمكين OpenFlow على عقد الشبكة، يصبح التحكم البرمجي التفصيلي ممكناً. لحسن الحظ، يمكن جلب OpenFlow إلى الشبكة بطريقة مقسمة حيث يمكن أن تعمل شبكة بنجاح كبير لفترة طويلة في وضع OpenFlow الهجين. يمكن أن يُطبق الوضع المختلط على الشبكة ككل، في وضع تكون فيه بعض العقد تدعم OpenFlow في حين أن بعضها الآخر لا. وعلاوة على ذلك، يمكن للعقد نفسها أن تعمل في وضع هجين، حيث يتم التحكم في بعض المنافذ أو شبكات VLAN بواسطة OpenFlow والبعض الآخر لا. لذا، يمكن تشغيل OpenFlow على مجموعة من المبدلات الطرفية. وبهذه الطريقة، يمكن اختبار التحكم المبرمج في الشبكة على مجموعة فرعية من المستخدمين. يوفر وضع التشغيل المختلط بيئة مناسبة لتحديد تطبيقات SDN التي تضيف قيمة حقيقية إليها.

2-3-3 المرحلة الثالثة: نشر نموذج SDN الأمثل

الانتقال من شبكة مختلطة إلى شبكة معرفة بالبرمجيات يجب أن يكون بشكل تدريجي جداً، وقد لا يكون في الواقع كذلك بشكل كامل. قد لا يؤدي نشر SDN إلا لتوفير فائدة متميزة في أقسام معينة من الشبكة، وعدم توسيعه إلى القطاعات الأخرى. من ناحية أخرى، في بعض الشبكات، تكون فوائد SDN واضحة للغاية بحيث يتم نشرها بسرعة على الشبكة بأكملها. أياً كان الحال، فإن التواجد السليم الغير متضارب للشبكات التقليدية والمعرفة بالبرمجيات يعني أن المستوى الأمثل من SDN في الشبكة يمكن اكتشافه بمرور الوقت، ومراجعتها بانتظام مع تطور التكنولوجيا.

إذاً يمكن أن يحدث الانتقال من شبكة تقليدية إلى شبكة معرفة بالبرمجيات ويجب أن يحدث بطريقة تدريجية. من المفترض أن تكون الهجرة إلى شبكة معرفة بالبرمجيات عملية ثلاثية المراحل، حيث كل خطوة يجب أن تعالج حالة للوصول إلى تحكم مبرمج في الشبكة، وبالتالي توفر الفائدة المرجوة منها. تتفادى عملية إدارة الهجرة بهذه الطريقة المشكلات وانقطاعات الشبكة المحتملة، وتسمح للشبكة بالحصول على المزايا الرئيسية لإدخال التحكم المبرمج إليها من تقليل التكاليف وزيادة المرونة وزيادة الأمان وتحسين تجربة المستخدم

2-4 صعوبات الانتقال إلى تقنية ال SDN :

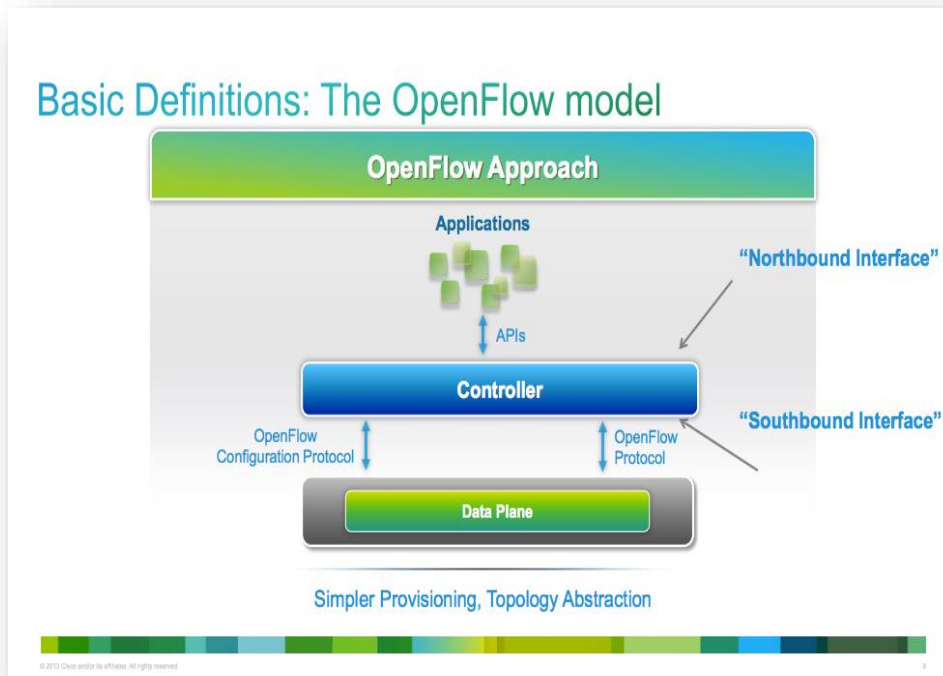
العقبة الكبرى التي تعيق التحول السريع إلى عالم SDN هي كيفية الانتقال من البنية التحتية الحالية بتصميمها وفكرها وطريقة إدارتها إلى الفكر الجديد

فهناك مجموعة من النقاط التي يجب أخذها بعين الاعتبار عند الانتقال إلى هذه التقنية:

1. القيام بتقسيم البنية التحتية للشبكات الحالية إلى عدة نطاقات لتبسيط عملية الانتقال فالتقسيم لنطاقات يجرأ التحديات التي ستقابلك.
2. التحضير للأدوات والأنظمة والطرق وحتى الأشخاص الذين سيساعدون للانتقال لهذه التقنية.
3. القيام بوضع خطة بسيطة وواضحة ودراسة تكلفة الانتقال المادية والزمنية.
4. القيام بعملية التوثيق لكل عملية سواء كانت صغيرة أم كبيرة.

2- 5 تعريف بروتوكول OpenFlow :

هو بروتوكول للتواصل بين طبقة التحكم وطبقة البنية التحتية والذي تم تطويره ولا يزال يتم تطويره من قبل منظمة ONF يعمل البروتوكول على الطبقة الثالثة ووظيفته نقل أوامر التوجيه وتحديد المسارات من وإلى الـ SDN Controller وهذه الأوامر تشمل إضافة مسار أو التعديل عليه أو حذف مسار وتعريف عام ينقل الـ Rules or Actions إلى مستوى الـ Data Plane ويقوم بعكس الوظيفة أيضا من خلال نقل أي متطلبات يحتاجها الـ Data Plane إلى مركز القيادة ونقصد بهذا الأمر الـ Unmatched Packet التي تصل إلى الراوتر ولا تملك مسار معروف.



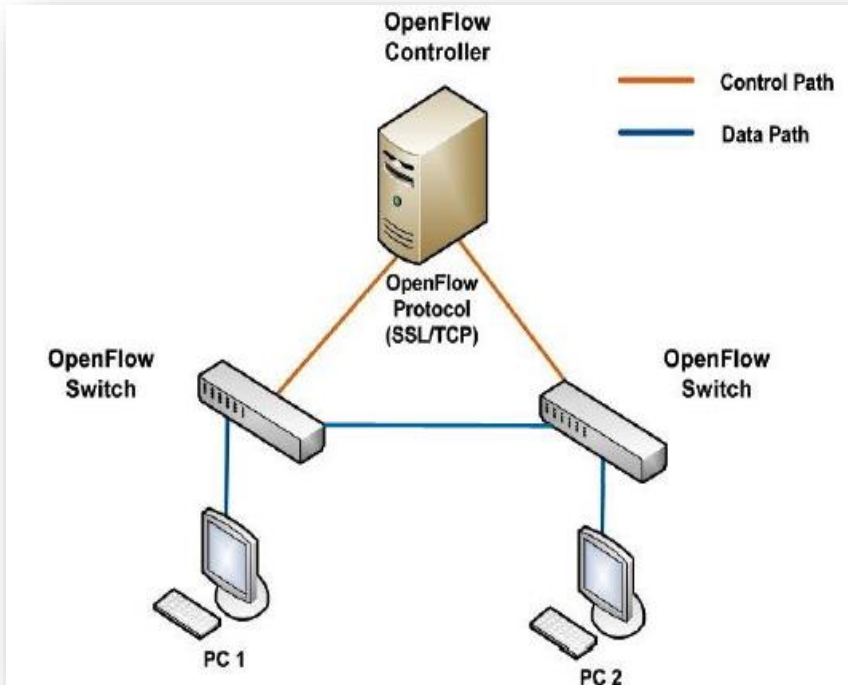
الشكل (1-2) OpenFlow

6-2 بنية شبكة OpenFlow :

تتكون بنية شبكة OpenFlow من ثلاثة مفاهيم أساسية:

1. المبدلات المتوافقة مع OpenFlow التي تشكل مستوى البيانات.
2. يتكون مستوى التحكم من واحد أو أكثر من وحدات تحكم OpenFlow
3. تربط قناة التحكم الآمنة المبدلات بمستوى التحكم.

تتواصل المبدلات مع المضيفين ومع بعضهم البعض باستخدام مسار البيانات يمكن للبرنامج توفيره ، وتتواصل وحدة التحكم مع المبدلات باستخدام مسار التحكم كما هو موضح في الشكل التالي:



الشكل (2-2) OpenFlow

7-2 إصدارات openflow :

إن الحاجة المتزايدة لخدمات openflow أدت الى وجود عدة إصدارات ونسخ منه لكي يدعم ويقدم

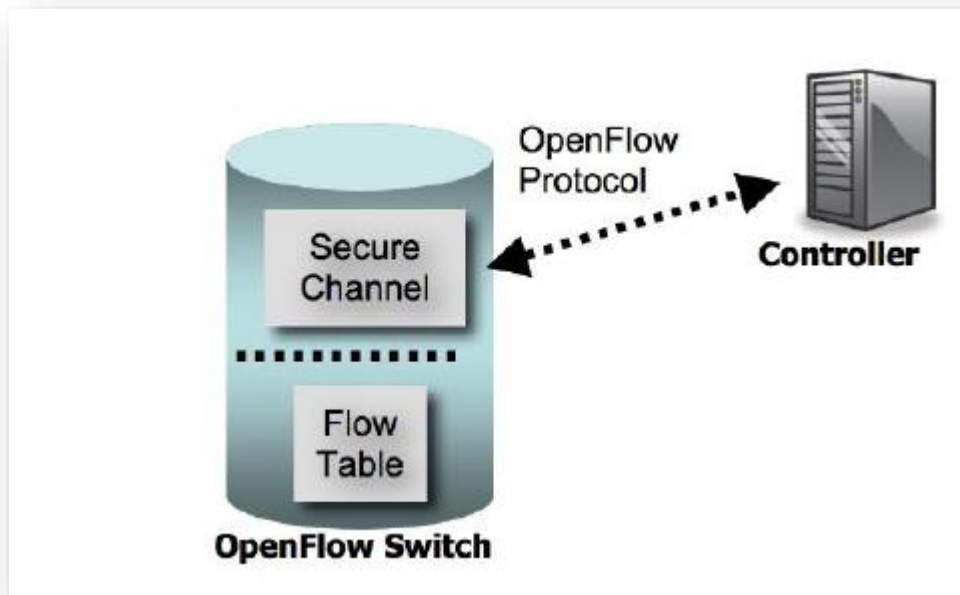
خدماته لكل المتحكمات حيث إن كل متحكم يتعامل مع نسخة خاصة به

المسار التاريخي لتطور openflow بدايةً بنسخة 1.1 الأقدم حتى نصل الى نسخة 1.5 اللاحث :

- 1.1 دعم MPLS , Q-in-Q , VLANs , والمسارات المتعددة , والجداول المتعددة والمنافذ المنطقية.
- 1.2 دعم لـ IPv6 والمطابقة القابلة للتوسيع
- 1.3 تجعل المعلومات بناءً على النوع والطول و يضيف إطار عمل التعبير عن القيمة (TLV) أيضاً دعماً لقياس التدفق بالتدفق ومعالجة رأس IPv6.
- 1.4 تركز الميزات الجديدة في openflow 1.4 على طبقة التحكم وتجعل openflow مستقر ومرن بدرجة كافية لمراكز البيانات ويساعد في النشر المتعدد لوحدة التحكم, و يمكن من مزامنة وحدات التحكم استناداً إلى عدد من العوامل , بما في ذلك مراقبة التدفق وإشعارات التغيير.
- 1.5 دعم لجداول الخروج من المبدل , وللقوائم المعرفة من قبل المستخدم , وميزات لها علاقة بجودة الخدمة , و يقدم تحسينات لتضمين جداول البث المتعدد التي تمكن مشغلي الشبكات من برمجة بيئة التبديل الخاصة بهم للتكامل بشكل أفضل مع تطبيقات SDN

2-8 OpenFlow enabled switch :

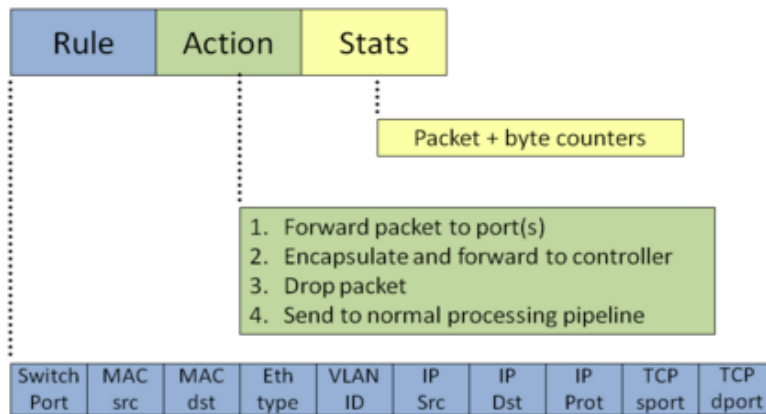
مبدل OpenFlow هو جهاز إعادة التوجيه الأساسي الذي يعيد توجيه الحزم وفقاً لجداول التدفق الخاص به ، والذي يشبه جداول إعادة التوجيه التقليدية ولكن لا تتم إدارته وصيانته بواسطة المحول. وهي متصلة بوحدة التحكم عبر قناة آمنة يتم من خلالها تبادل رسائل OpenFlow بين المحول ووحدة التحكم كما هو موضح في الشكل التالي :



الشكل (2-3) OpenFlow

Flow Table 8-2 :

يحتوي المبدل في شبكة OpenFlow على جدول تدفق واحد (أو أكثر) يحتوي على مجموعة من الإدخالات ، يتكون كل منها من الحقول كما هو موضح في الشكل (3-4). تتم مقارنة جميع الحزم التي تمت معالجتها بواسطة المبدل مقابل جدول التدفق. إذا تطابق رأس الحزمة مع إدخال تدفق ، يتم تنفيذ إجراء لهذا الإدخال على الحزمة (على سبيل المثال ، قد يكون الإجراء هو إعادة توجيه حزمة إلى منفذ محدد). إذا لم يتم العثور على تطابق ، يتم إعادة توجيه الحزمة إلى وحدة التحكم عبر القناة الآمنة. العدادات محجوزة لجمع الإحصائيات حول التدفقات. يقومون بتخزين عدد الحزم والبايت المستلمة ، بالإضافة إلى مدة التدفق.



الشكل (4-2) Data Flow Table Entry

1-8-2 يوجد ثلاثة أقسام رئيسية للـ Data Flow Table Entry:

Rule: وهو كما تلاحظوا يتألف من قسمين الـ Layer 2 والـ Layer 3 والتي سوف نبني عليها المسارات ونحدد من أين وإلى أين سوف تتم عملية التوجيه.

Action: وهو الأمر الذي يجب تنفيذه كتمرير الباكيث أو تجاهلها أو إرسالها إلى المتحكم من خلال الـ Data Plane .

Stats: الإحصائيات وهي مهمة لمتابعة آخر المستجديات ووضع الـ Dead Line للمدخلات الموجودة على الـ Table (اي لو في حال لم يتم عمل اي Match لباكيث منذ فترة طويلة فسوف يتم حذفها من الـ Flow Table)

2-8-2 الإجراءات المتخذة بشأن التدفقات

إذا كانت حزمة الإدخال تتطابق مع أحد حقول المطابقة في جدول التدفق ، فيجب أن يشير الإجراء إلى تلك الحزمة. يجب أن يتم دعم إعادة توجيه إجراء الحزم إلى كل منفذ فعلي بواسطة مبدل OpenFlow. بالإضافة إلى ذلك ، هناك منافذ افتراضية يحددها معيار OpenFlow كأهداف خاصة يمكن إعادة توجيه الحزم إليها ، مثل ماهو موضح في الجدول التالي. هذه الإجراءات هي إجراءات "مطلوبة" وإجراءات "اختيارية". يجب أن تدعم جميع المبدلات الإجراءات المطلوبة لتكون متوافقة مع OpenFlow والإجراءات الاختيارية التي أثبتت أنها إجراءات مفيدة ولكنها ليست مدعومة بالضرورة في مبدل متوافق مع OpenFlow.

قائمة المنافذ الافتراضية للإجراءات "المطلوبة":

Virtual Port	Description
ALL	Forward the packet to all ports except the received port
CONTROLLER	Encapsulate the packet and send it to the controller
LOCAL	Send the packet to the local networking stack for the switch
TABLE	Perform actions described in the flow table. Only for packet-out messages
IN_PORT	Send the packet to the received port

الجدول (1-2) قائمة المنافذ الافتراضية للإجراءات "المطلوبة"

قائمة المنافذ الافتراضية للإجراءات " اختيارية ":

Virtual Port	Description
NORMAL	Forward the packet using the traditional forwarding methods, i.e. traditional L2, VLAN, and L3 processing.
FLOOD	Send the packet along the minimum spanning tree, not including the incoming interface. Each port in the OpenFlow enabled switch has a NO_FLOOD-bit, which indicates that the port doesn't belong to the minimum spanning tree. The packets that match that flow entry are forwarded to the ports that have a NO_FLOOD-bit.

الجدول (2-2) قائمة المنافذ الافتراضية للإجراءات " اختيارية "

بجانب إجراء إعادة التوجيه ، توجد إجراءات أخرى في جدول التدفق:

DROP: إجراء مطلوب يشار إليه بقائمة إجراءات فارغة. يتم إسقاط جميع الحزم المتطابقة مع قائمة الإجراءات الفارغة.

قائمة الانتظار: يمكن استخدام هذا الإجراء الاختياري لوضع الحزم في قائمة انتظار مرتبطة بمنفذ من أجل توفير جودة الخدمة.

تعديل الحقل: يتم استخدام هذا الإجراء الاختياري لتعديل حقل رأس معين للحزمة الواردة. يمكن إجراء التعديل التالي:

- تعيين معرف VLAN والأولوية.
- شريط رأس VLAN.
- تعديل مصدر Ethernet وعنوان MAC الوجهة.
- تعديل مصدر IP والوجهة.
- تعديل بتات IP TOS
- تعديل مصدر طبقة النقل ومنفذ الوجهة

2-8-3 أنواع التدفقات

يمكن تصنيف التدفقات المأهولة من وحدة تحكم OpenFlow إلى أنواع: التدفقات الصغيرة والتدفقات المجمعة .

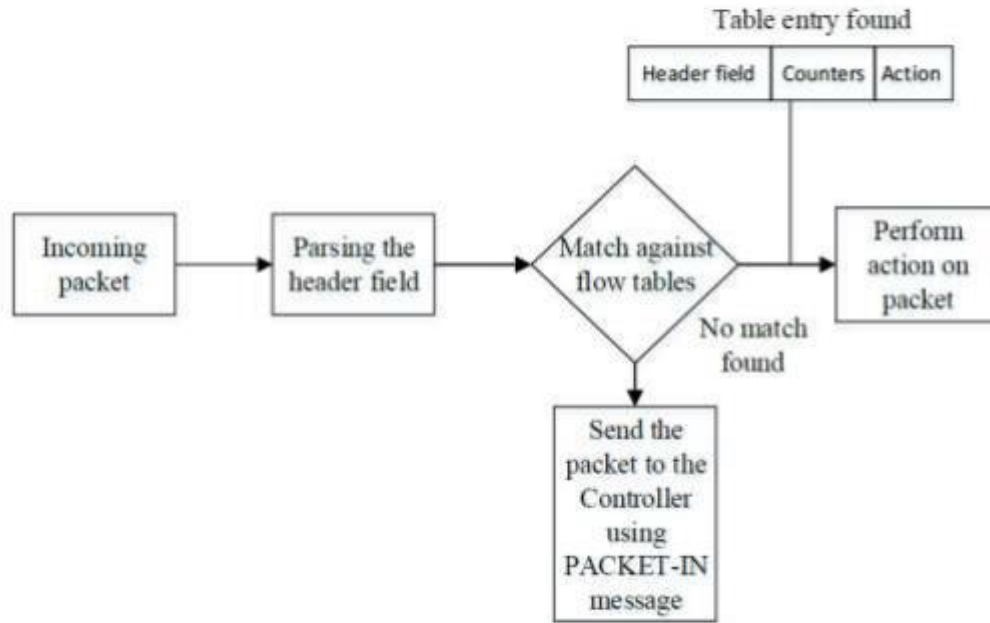
- **Microflows**: هذا النوع من التدفق مفيد عندما يلزم تثبيت عدد صغير من التدفقات في المبدل ، على سبيل المثال شبكة الحرم الجامعي. تحتوي جداول التدفق في هذا النوع على إدخال واحد لكل تدفق والمطابقة الدقيقة مطلوبة لتنفيذ إجراء ما.
- **مُجمّع**: مفيد للشبكات الكبيرة التي تتطلب عددًا كبيرًا من إدخلات جدول التدفق ، على سبيل المثال شبكات العمود الفقري. في هذا النوع ، يغطي إدخال تدفق واحد (Wildcarded) عددًا كبيرًا من التدفقات ، يجب أن ينتمي كل منها إلى فئة معينة.

يمكن تصنيف كل نوع من الأنواع المذكورة أعلاه إلى تدفقات تفاعلية واستباقية.

- **تفاعلي**: في هذا النوع ، تكون وحدة التحكم في وضع الخمول حتى تستقبل الحزمة الأولى من مفتاح OpenFlow. تحلل وحدة التحكم الحزمة الواردة ثم تُدرج إدخال تدفق جديد في جدول تدفق المحول. في هذا النوع ، يحتاج كل إدخال تدفق جديد إلى وقت إعداد إضافي صغير. إذا فشل الاتصال بين وحدة التحكم والمحول ، ولم يكن للمحول القدرة على إعادة توجيه الحزمة كمفتاح تقليدي ، فلن يتمكن من إعادة توجيه الحزم إلى المضيفين.
- **استباقي**: في هذا النوع ، تقوم وحدة التحكم بالتثبيت المسبق لإدخلات التدفق في جدول تدفق المحول دون الحاجة إلى استلام الحزمة الأولى من التدفق. لا يتطلب وقتًا إضافيًا لإعداد التدفق وفي حالة فشل الارتباط بين وحدة التحكم والمفتاح ، فلن يتم تعطيل حركة المرور.

2-9 آلية إعادة توجيه الحزمة في شبكة OpenFlow :

عندما يسقبل المحول حزمة ، فإنه يوزع حقل الرأس ويتحقق منه مقابل القواعد الموجودة في جدول التدفق. في حالة وجود تطابق ، يتم أخذ إجراء في جدول التدفق في الاعتبار. إذا تم العثور على العديد من التطابقات ، تتم مطابقة الحزم مع إدخال تدفق معين بناءً على تحديد الأولويات ، أي تحديد إدخال التدفق ذي الأولوية القصوى. ثم يقوم المبدل بتحديث عدادات إدخال جدول التدفق هذا. أخيرًا ، يقوم المفتاح بإعادة توجيه الحزمة إلى منفذ. إذا لم تتطابق الحزمة الواردة مع أي إدخال تدفق في جدول التدفق ، فسيقوم المبدل بإعادة توجيه الحزمة إلى وحدة التحكم لحساب أي منطق يجب تضمينه في هذه الحزمة والحزم المستقبلية المماثلة. يمكن توضيح عملية آلية إعادة توجيه الحزمة في الشكل للمخطط الانسيابي.



الشكل (2-5) رسائل بين المتحكم والمبدل

2-10 نلخص عمل البروتوكول OF :

أنه عندما يصل تدفق جديد إلى المبدل تتم مطابقة هذا التدفق مع مجموعة القواعد أو المداخل الموجودة في جدول التدفق , ومن ثم يتم التعامل مع هذا التدفق وفق الحالات التالية:

- في حال حدوث مطابقة يتم التصرف بالرمزة وفقاً للقاعدة الموجودة وتعديل البيانات الموجودة في عمود الإحصائيات .
- في حال عدم وجود تطابق يتم إرسال هذه الرزمة بشكل كامل أو ترويسها إلى المتحكم وذلك لاتخاذ القرار المناسب .
- بعد أن يتم إتخاذ القرار , يقوم المتحكم بإرسال هذا القرار إلى المبدل ويقوم بتحديث جدول التدفق الخاص به
- تتم هذه الآلية وفق تبادل عدة رسائل بين المتحكم والمبدل , والتي سيتم توضيحها

2-11 رسائل الاتصال OpenFlow :

توجد ثلاث فئات من الاتصال في بروتوكول OpenFlow:

1. Controller-to-switch أي بين المتحكم والمبدل.

2. غير متزامن.

3. متزامن.

2-11-1: Controller-to-switch رسائل

رسائل التي يتم تبادلها بين المتحكم والمبدل يتم إنشاؤها من قبل المتحكم وترسل إلى المبدل.

يوجد عدة أنواع مختلفة من هذه الرسائل:

- Features: هذه الرسائل هي وضع request/reply حيث يرسل المتحكم رسالة طلب , الميزات للمبدل , الذي بدوره يرسل ميزات ويحدد القدرات التي يدعمها المبدل.
- Configuration: تمكن هذه الرسالة المتحكم من تعيين بارامترات التهيئة واستعلامها في المبدل , يقوم المبدل بالرد على الاستعلام و يرسل المعلومات المطلوبة إلى المتحكم.
- Modify-Stat: يتم استخدامه لإضافة / حذف أو تعديل التدفقات في جداول التدفق ولضبط خصائص منفذ المبدل.
- Read-State : هذه الرسائل تمكن المتحكم من جمع إحصاءات من يقوم بتحويل جداول التدفق والمنافذ وإدخالات التدفق الفردية
- Send-Packet: يستخدم المتحكم هذه الرسائل لإرسال الحزم الى منفذ محدد على مبدل.
- Barrier : هذه الرسائل هي صيغة request/reply حيث يكون طلب Barrier أرسل من قبل المتحكم لضمان وصول الرسائل السابقة. يقوم المبدل بإرسال رد من Barrier عند التنفيذ الناجح للرسائل.

2-11-2: Asynchronous Messages

تبدأ هذه الرسائل من قبل المبدل وترسل إلى المتحكم .هنالك أربعة أنواع من الرسائل الرئيسية غير المتزامنة:

- Packet-in: تحتوي رسالة packet-in على حزمة يتم إرسالها إلى المتحكم . إما بسبب عدم وجود تطابق لأي إدخال تدفق في جدول التدفق Flow table أو أنه يطابق إجراء يأمر المبدل بإرساله إلى المتحكم. إذا كان لدى المبدل إمكانيات تخزين الحزمة , فإنه يرسل فقط جزءاً من حقول الرأس (بشكل افتراضي 128

بايت) ويخزن الباقي بمعرف مخزن مؤقت ليتم استخدامه بواسطة المتحكم للتبرع بإدخال التدفق الجديد مع ذلك المخزن المؤقت بايت

- Flow-Removed : تتم إضافة مدخلات على جدول التدفق وفق قيم زمنية مهلة والوقت الثابت عن طريق رسالة تعديل تدفق المتحكم . تشير مهلة الحمول إلى عدم وجود نشاط للتدفق , وتشير مهلة العمل إلى عمر التدفق . بعد انتهاء هذه القيم , تتم إزالة التدفقات تلقائيًا , ويتم إخبار المتحكم بذلك من خلال رسالة

FLOW_REMOVED

- Port-status: يتم إرسال هذه الرسالة إلى المتحكم إذا قام منفذ التبديل بتغيير حالته , على سبيل المثال إذا كانت تنخفض لسبب معين
- Error : إذا حدث خطأ في أحد المبدلات , فيمكن إخطار المتحكم عن طريق هذا النوع من الرسائل

3-11-2: Symmetric Messages

تتم تهيئة الرسائل المتماثلة من خلال كل من المحول والمتحكم في أي اتجاه . وفق الموصفات

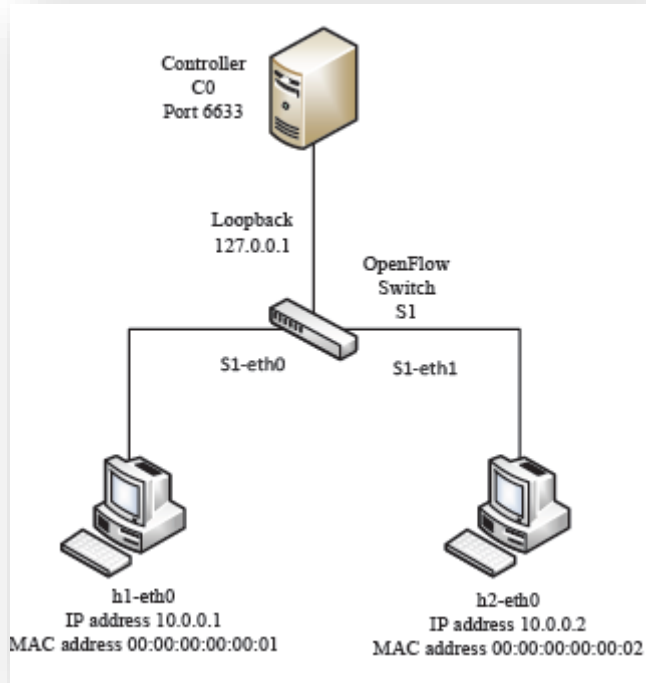
هذه الرسائل هي OpenFlow :

- Hello: يتم تبادل هذه الرسائل بين المحول والمتحكم عند بدء الاتصال .
- Echo : هذه الرسائل موجودة في نموذج الطلب / الرد , حيث يمكن إرسالها إما عن طريق مفتاح تبديل أو وحدة تحكم , ويجب أن تعرض latency و / أو النطاق الترددي , و / أو رسائل , التواصل من وصلة المتحكم.
- Vendor : يقوم كل من المبدل والمتحكم من خلال هذه الرسالة بتسمية المنتج الخاص به Vendor.

2-12: شرح توضيحي للرسائل المتبادلة في شبكة OpenFlow :

من أجل توضيح الرسائل السابقة في شبكة OpenFlow الحقيقية , استخدمنا محاكي شبكة Mininet لمحاكاة مضيفين متصلين بمحول ووحدة تحكم . نشرح أولاً إنشاء اتصال

Switch- Controller , ثم الاتصال من المضيف إلى المضيف من خلال مفتاح OpenFlow ووحدة التحكم.



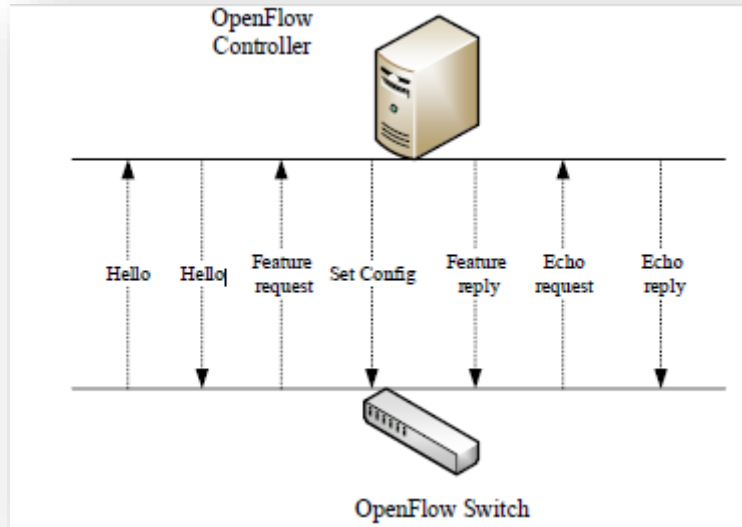
الشكل (6-2) Symmetric Messages

2-12-1 الرسائل بين المبدل ووحدة التحكم:

عندما يتصل المبدل بشبكة OpenFlow ، فإنه ينشئ اتصال TCP بعنوان IP الخاص بوحدة التحكم (واجهة Loopback 127.0.0.1) ومنفذ افتراضي 6633.

بعد هذه العملية ، يبدأ كلا الجانبين في تبادل رسائل Hello التي تتضمن أعلى إصدار من OpenFlow . بعد ذلك ، يتم إرسال رسالة طلب الميزة **Feature request** بواسطة وحدة التحكم لمعرفة المنافذ المتاحة في المبدل ، والذي بدوره يرد برسالة رد الميزة **Feature reply** التي تحتوي على قائمة المنافذ وسرعة المنافذ والجداول والإجراءات المدعومة. يتم إرسال رسالة **Set config** بعد ذلك بواسطة وحدة التحكم لمطالبة المفتاح بإرسال انتهاء صلاحية التدفق.

أخيراً ، يتم إرسال **echo request, echo reply** بشكل متكرر بين المبدل ووحدة التحكم لتبادل المعلومات المتعلقة بالنطاق الترددي ووقت الاستجابة وحيوية اتصالاتهما. توضح الأشكال أدناه تسلسل المعالجة والتقاط Wireshark للحزم ،



الشكل (2-7) الرسائل بين المبدل والمتحكم

2-12-2 تبادل الرسائل بين مضيفين :

لتوضيح كيفية إجراء اتصال مضيف إلى مضيف في شبكة OpenFlow ، نستخدم أداة Ping لإرسال حزم ICMP من h1 إلى h2 والعكس بالعكس. تبدأ العملية عندما يرسل h1 طلب ARP إلى المبدل ، ويسأل عن عنوان MAC h2 ، ولا يعرف المبدل كيفية التعامل مع الحزمة ، لذلك يرسل الحزمة كرسالة PACKET-IN إلى وحدة التحكم. تجيب وحدة التحكم برسالة PACKETOUT تحتوي على إجراء يوجه المبدل إلى إرسال الحزمة إلى جميع المنافذ باستثناء المنفذ الوارد وانتظر الرد على هذا الطلب. عندما ترد h2 على هذا الطلب ، يرسل المبدل أيضًا هذا الرد إلى وحدة التحكم لأنه ليس لديه فكرة عن مكان إعادة توجيه هذه الحزمة. عندما تتلقى وحدة التحكم رد ARP ، فإنها ترسل رسالة FLOW-MOD لتثبيت إدخال تدفق جديد لردود ARP المستقبلية من h2 البعيد إلى h1 ليتم إعادة توجيهها مباشرة بواسطة المبدل دون إخطار وحدة التحكم. تحدث نفس العملية عندما ترسل h1 طلب / رد ICMP ، وعندما ترسل h2 طلب ARP لحل عنوان H1 MAC ورد ARP الناتج. في النهاية ، سيتم تثبيت خمسة مداخل تدفق جديدة في جدول تدفق المحول بواسطة وحدة التحكم OpenFlow

الفصل الثالث

المتحكمات في الشبكات المعرفة بالبرمجيات

3-1 مقدمة:

تختلف طبولوجيا الشبكة في تقنية الشبكات المعرفة بالبرمجيات كما هو الحال في الشبكات التقليدية حيث أن طبولوجيا الشبكة هي ترتيب العقد في الشبكة بطريقة معينة. قد تشمل العقد المستخدمين النهائيين (أو الأجهزة المضيفة)، وأجهزة الشبكة الأساسية مثل المبدلات، وأجهزة التوجيه، إلخ. وقد تشمل أيضاً الطابعات والماسحات الضوئية وأجهزة الفاكس وما إلى ذلك.

طبولوجيا شبكة OpenFlow ثلاثة هي:

1. Single

2. Linear

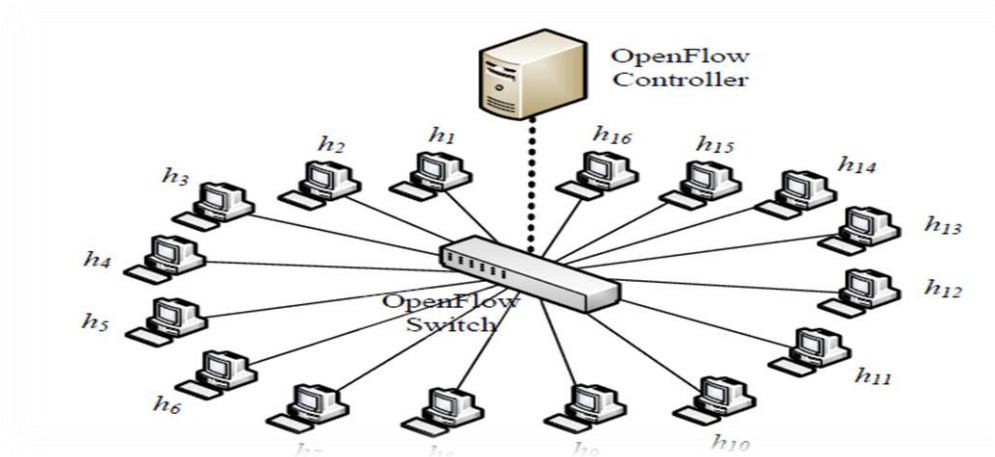
3. Tree

3-2 Single topology :

تتكون من مبدل واحد يدعم OpenFlow يتصل بالعديد من المضيفين. بدوره المبدل متصل مع وحدة تحكم OpenFlow عبر قناة آمنة.

التعليمة التي يتم استخدامها في المحاكى mininet للحصول على هذه ال topology:

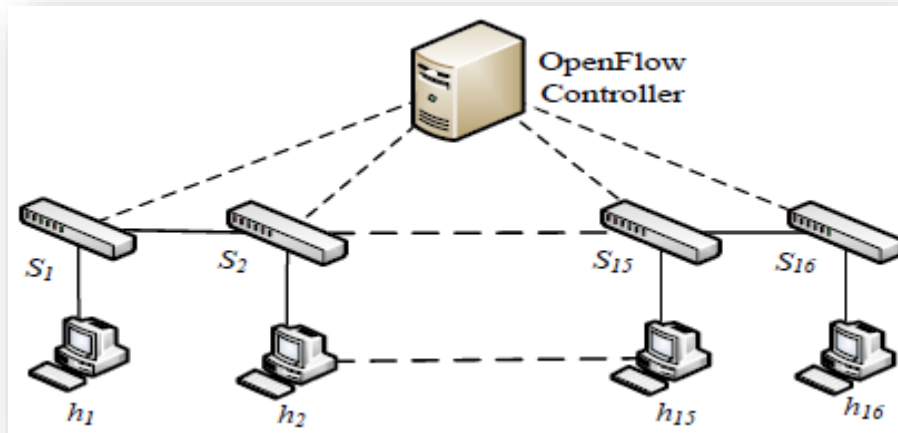
```
#sudo mn --topo=single,16 --controller remote.
```



الشكل (1-3) Single topology

: Linear topology 3-3

في الطوبولوجيا الخطية ، إذا كان هناك N مضيف في الشبكة ، فيجب طلب N من المبدلات. هذا يعني أن كل مضيف سوف يتصل مع مبدله الخاص. على سبيل المثال ، سيتصل المضيف H1 مع المبدل S1 ، والمضيف H2 مع المبدل S2 ، وجميع المبدلات متصلة مع بعضها البعض والتي بدورها متصلة مع وحدة تحكم مشتركة.



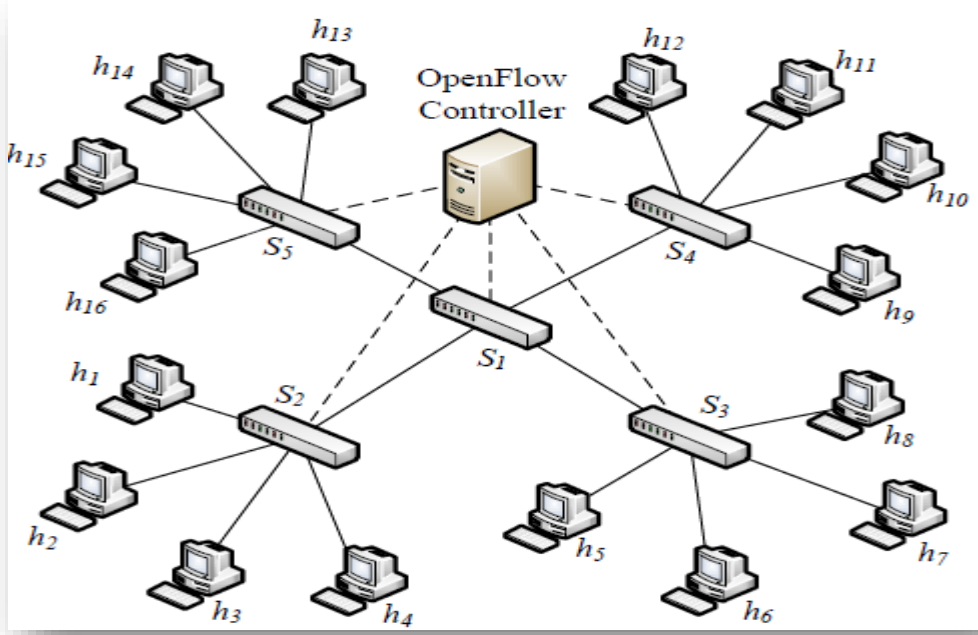
الشكل (2-3) linear topology

التعليمة التي يتم استخدامها في ال mininet للحصول على هذه ال topology:

```
#sudo mn -- topo=linear,16 --controller remote.
```

: tree topology 4-3

تعتمد على ترتيب المبدلات والمضيفات بطريقة شجرة. أي أن فروع متعددة موجودة في الطوبولوجيا وفي هذه الفروع ترتبط المبدلات والمضيفات المتعددة وفقاً لتصميم الطوبولوجيا



الشكل (3-3) tree topology

التعليمة التي يتم استخدامها في ال mininet للحصول على هذه ال **topology**:

#sudo mn --topo tree,depth=2, fanout=4 --controller remote.

3-5 المتحكمات في الشبكات المعرفة بالبرمجيات:

العديد من المتحكمات قد تم تطويرها منذ أن نشأت فكرة SDN إن اختيار المتحكم المناسب والذي يلبي متطلبات الشبكة هو تحدي بذاته حيث يجب أن يتميز المتحكم بمعدل إنتاجية **Throughput** عال وزمن استجابة منخفض بالإضافة إلى تأمين الحماية من طرود المتطفلين وتأمين استقرار الشبكة. تكمن أهمية اختيار متحكمات مناسبة للعمل في الشبكات المعرفة بالبرمجيات وبالتالي زيادة كفاءة هذه الشبكات وزيادة فعاليتها .

التوفير في تكاليف بناء وإدارة أجهزة البنية التحتية للشبكات هي أهم مزايا الشبكات المعرفة بالبرمجيات كما تقدم تحكم وإدارة مركزية وشبكة قابلة للبرمجة مما يتيح المرونة في التصميم والإدارة وتحسين في أمن ووثوقية الشبكة. إن المتحكم في الشبكات المعرفة بالبرمجيات مسؤول عن المحافظة على استقرار الشبكة وتطبيق السياسات والقواعد في الشبكة بالإضافة إلى توزيع التعليمات إلى أجهزة الشبكة المختلفة. فالمتحكم يملك نظرة عامة منطقية عن كل مكونات الشبكة. سنتعرف على أبرز أنواع هذه المتحكمات.

3-5-1 المتحكم POX في SDN :

يعتبر المتحكم العنصر الأكثر أهمية في SDN حيث يمثل عقل الشبكة , ويقوم بإصدار القواعد التي تنفذها الأجهزة الموجودة في طبقة (Data Plane) ويتواصل مع هذه الطبقة بواسطة البروتوكول (Open Flow), بينما يتواصل مع طبقة التطبيقات عن طريق ما يسمى بالجسر الشمالي (Northbound Interface) .

يعتبر المتحكم POX من المتحكمات الهامة والشهيرة في SDN , وقد تمت برمجته بلغة (Python), كما أنه يوفر منصة برمجية سهلة ومرنة لكتابة التطبيقات التي يمكن تنفيذها في الشبكة.

تتناول أغلب الأبحاث في مجال 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.

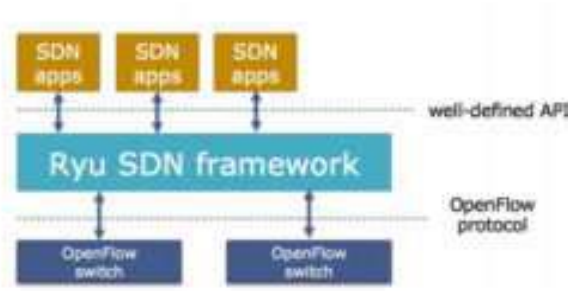
3-5-2 المتحكم NOX :

أول بروتوكول تم تطويره في سنة 2008 جنباً إلى جنب مع بروتوكول OpenFlow وقد كان الأساس لتطوير العديد من المتحكمات التي جاءت بعده. يقدم هذا المتحكم واجهة برمجية تطبيقات API مكتوبة بلغة ++C وهو يدعم الإصدار 1.0 من بروتوكول OpenFlow . يتضمن مكونات بسيطة لاكتشاف الشبكة والتعامل مع مبدلات OpenFlow وهو لا يدعم تعدد النياسب .

تقدم نواة NOX وظائف مساعدة كمعالجة الرزم وإدارة الأحداث المختلفة بالإضافة إلى واجهة التخاطب مع مبدلات OpenFlow وعمليات الإدخال والإخراج I/O . وتتضمن الطبقة الوسطى المكونات المبنية ضمناً وهي مدير الاتصال - مدير OpenFlow - مدير الأحداث. يمثل نظام الأحداث أحداثاً عالية المستوى ومنخفضة المستوى في الشبكة, هذه الأحداث تقدّم معلومات معينة وآلية معالجتها. تتضمن هذه الأحداث ما يلي : اتصال مبدل جديد - حذف مبدل - وصول رزمة جديدة - إضافة جدول إلى مدخل التوجيه ...

3-5-3 المتحكم RYU :

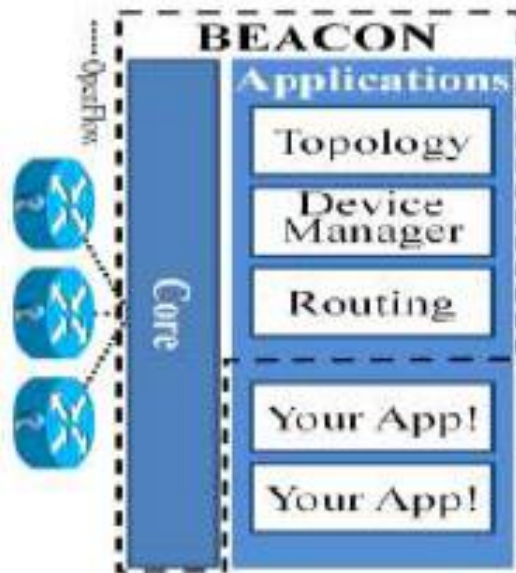
هو متحكم مفتوح المصدر مكتوب بلغة البايثون, يقدم مكونات معرفّة بشكل جيد تمكن المطورين من كتابة تطبيقات عديدة للشبكة. يدعم RYU معظم إصدارات بروتوكول OpenFlow من 1.0 وحتى 1.5. يدعم بروتوكول STP وبذلك يمنع حدوث الحلقات في الشبكة. وهو لا يدعم تعدد النياسب (Multithreading) في عمله. يبين الشكل التالي هيكلية المتحكم RYU



الشكل (3-4) هيكلية المتحكم RYU

3-5-4 المتحكم Beacon :

المتحكم Beacon هو متحكم مفتوح المصدر مبني بلغة جافا ويدعم بروتوكول OpenFlow في عمله. تم إنشاؤه عام 2010 وهو مستخدم بشكل كبير في الأبحاث والتعليم وكان الأساس للمتحكم FloodLight والعمليات المقادة بالأحداث بالإضافة إلى النمطية وتعدد النياسب. يقدم منصة للتحكم بأجهزة الشبكة باستخدام بروتوكول OpenFlow بالإضافة لمجموعة من التطبيقات المبنية داخلياً والتي تؤمن وظائف مستوى التحكم الرئيسية. يمكن له أن يعمل على أية منصة تشغيل بالإضافة إلى أنه ديناميكي أي يمكن تشغيل وإيقاف وإعادة تشغيل أية خدمة فيه دون الحاجة إلى إعادة تشغيله. يبين الشكل هيكلية المتحكم



الشكل (3-5) هيكلية المتحكم Beacon

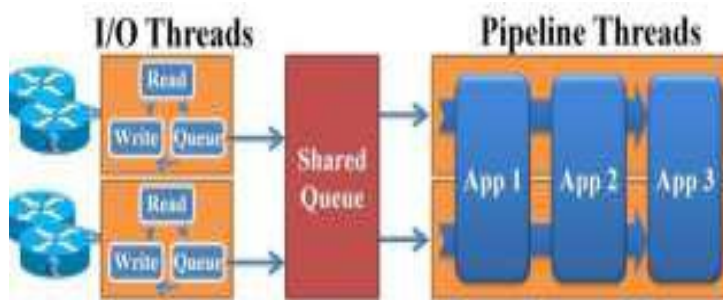
يوفر Beacon أداءً عالياً جداً من خلال دعمه لتعدد النياسب من خلال طريقتين : الرتل المشترك - Run To Completion

- الرتل المشترك :

يتكون من مجموعتين من النياسب. أول مجموعة من النياسب هي نياسب I/O تقوم بقراءة رسائل OpenFlow من المبدلات ثم وضعها في رتل مشترك. يتم اسناد كل مبدل لنيسب إدخال / إخراج واحد, يمكن إسناد عدة مبدلات لنفس النيسب. يقوم هذا النيسب أيضاً بكتابة رسائل OpenFlow الخارجة لمبدلاتها.

المجموعة الثانية من النياسب, تقوم بإخراج الرسائل من الرتل المشترك وتنفيذ كل رسالة خلال خط معالجة يعود إلى نوع معين من الرسائل. هذا التصميم يستلزم وجود قفل على الرتل لمشاركته. وهو متغير يمكن أن يمتلك عدة أرتال, ربما واحد لكل مبدل او حتى أكثر من واحد لكل مبدل وكل رتل بخصائص مختلفة. هذا يؤدي إلى تحسين وزيادة معالجة الرسائل بين المبدلات من خلال خوارزمية

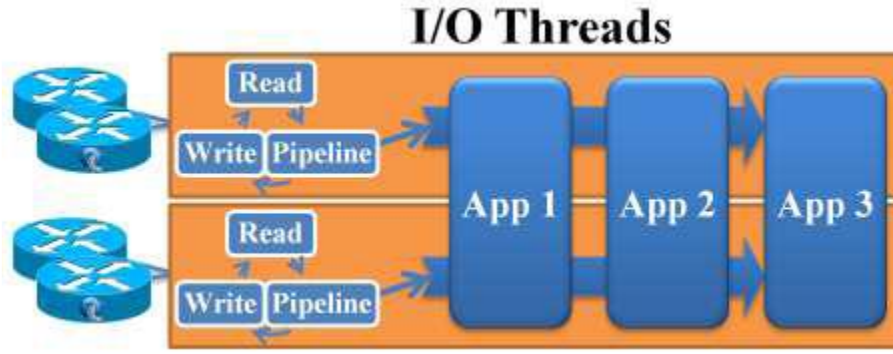
round-robin التي تخدم الرتل. الشكل (3-4) يوضح آلية الرتل المشترك.



الشكل (3-6) آلية الرتل المشترك

- Run To completion :

تملك مجموعة واحدة من النياسب. كل نيسب يشبه في عمله نياسب I/O في الرتل المشترك ولكن عوضاً عن وضع الرسائل في رتل ليتم تنفيذها بخطوط النياسب, فإنه يقوم بقراءة الرسالة كاملة حتى الإنتهاء منها والانتقال إلى الرسالة التالية. هذه الطريقة لا تحتاج إلى قفل. يتم تخصيص نيسب لكل مبدل يقوم بقراءة الرسائل ومعالجتها. كما في الشكل (3-5):



الشكل (7-3) Run To completion

يعاني هذا التصميم من قصور، حيث أنه من الممكن أن تكون أحد النياصب مشغولة بعدد كبير من الرسائل في حين توجد نياصب أخرى في حالة Idle. يستخدم Beacon هذا النوع من المعالجة عندما يكون عدد النياصب محدداً مسبقاً.

3-5-5 المتحكم Floodlight:

متحكم مقدم من قبل شركة Big switch يدعم بروتوكول OpenFlow بمختلف إصداراته. وهو مفتوح المصدر مبني بلغة جافا. ويقدم واجهة برمجية تطبيقات REST API متقدمة تساعد على التحكم بكل وظائف المتحكم وعرض قيم بارامتراته في صفحة ويب. عند تشغيل المتحكم يتم معه تشغيل العديد من التطبيقات المكتوبة بلغة الجافا. تعمل الخدمات المقدمة من قبل الواجهة الشمالية REST API على المنفذ 8080 ويمكن لأي تطبيق مكتوب بأية لغة برمجة النفاذ إلى معلومات المتحكم وعرضها وتنفيذ أوامر معينة.

يتضمن المتحكم العديد من الوحدات البرمجية المختلفة التي تلي متطلبات شبكة SDN مثل اكتشاف الشبكة – إدارة الأجهزة – مدير طوبولوجيا الشبكة – اكتشاف الوصلات – التوجيه وغيرها .. تتصل هذه المكونات مع بعضها عبر واجهات interface برمجية تقوم بتحقيقها.

عندما يقوم مضيف ما في الشبكة بإرسال طرد ما إلى مضيف آخر، فإن هذه الرزمة أولاً تصل إلى المبدل، يبحث المبدل في جدول التدفق الخاص به Flow Table عن وجود تطابق، في حال عدم وجود التطابق تُرسل الرزمة إلى المتحكم على شكل رزمة packet_in ليقرر بعدها المتحكم وجهة الرزمة وإعلام المبدل بذلك الذي يقوم بإضافة مدخل Flow Entry إلى جدول التدفق الخاص به.

3-6 اختبار أداء المتحكمات :

تم اختبار أداء المتحكمات بواسطة مخدمين متصلين بوساطة وصلة بسرعة GB10

المخدم الأول يستخدم لتشغيل المتحكم ، والمخدم الثاني يستخدم لتوليد ال Traffic .

تم إجراء الاختبار على سبعة متحكمات:

1 (NOX : يدعم multi-threaded ، مبني بلغة ++C بالاعتماد على

المكتبة Boost .

2 (POX : لا يدعم multi-threaded ، مبني بلغة بايثون ويستخدم بشكل

واسع من قبل الباحثين.

3 (Beacon : يدعم multi-threaded ومبني بلغة java بالاعتماد على

منصة spring و OSGI .

4 (Floodlight : يدعم multi-threaded ومبني بلغة java بالاعتماد على

منصة Netty .

5 (MUL : يدعم multi-threaded ومبني بلغة C .

6 (Maestro : يدعم multi-threaded ومبني بلغة java بالاعتماد على

مكتبة java.nio .

7 (Ryu : لا يدعم multi-threaded ، مبني بلغة بايثون ويدعم أغلب نسخ

بوتوكول OpenFlow .

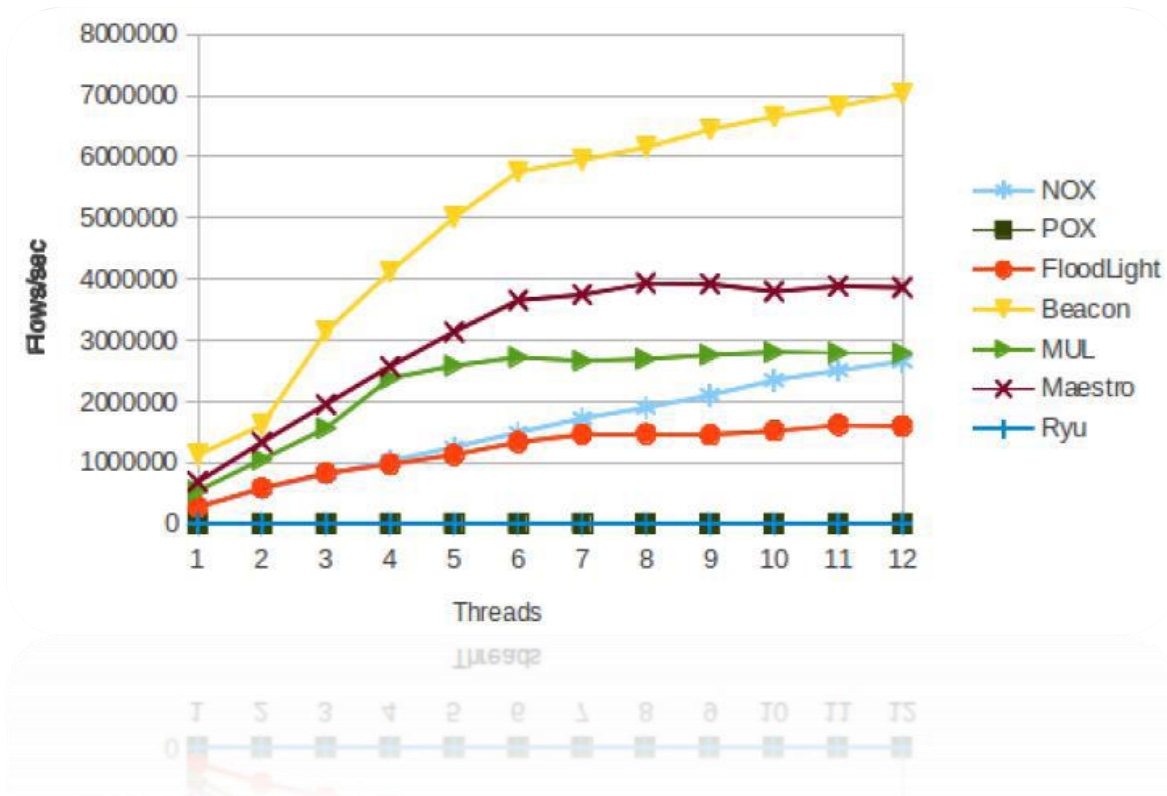
طبق على كل متحكم أداة L2 switching learning ، وتم القياس باستخدام الأدوات cbench و

hcrprobe .

الأداة **cbench** : تمكنا من قياس كفاءة جهاز المتحكم كأصغر و أقصى زمن استجابة، وقياس الانتاجية.

الأداة **hcrprobe** : يحدد نموذج لتوليد رسائل OpenFlow ، وعدد المرات لمحاولات إعادة الاتصال في

حال أن المتحكم أغلق الاتصالات فجأة .



الشكل (8-3) مقارنة متحكمات

نلاحظ أن المتحكمات POX, Ryu التي لا تدعم multi-threaded إنتاجيتها ثابتة مع ازدياد عدد النيباسب.

وإن المتحكم Beacon يعطي أفضل إنتاجية مع زيادة عدد النيباسب وذلك لاستخدامه أرتال مشتركة للرسائل القادمة.

زمن الاستجابة الوسطي لكل المتحكمات يتراوح بين ال (80-100) ms ، وعند الاختبار لمدة طويلة لوحظ أن حصل فقدان في الاتصالات بين المبدلات والمتحكم.

الفصل الرابع

القسم العملي

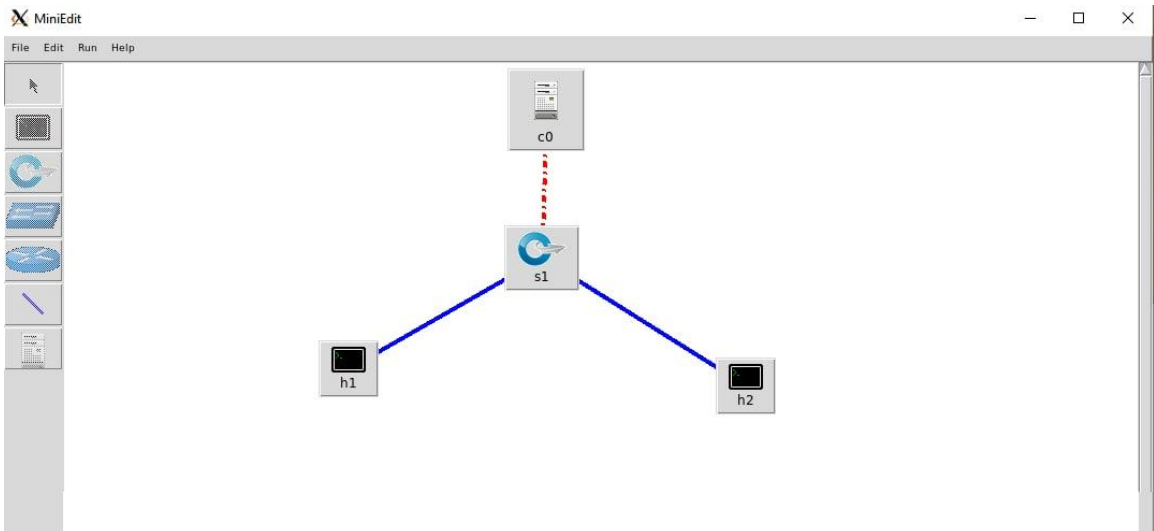
1-4 مقدمة حول أداة ال mininet :

هي عبارة عن محاكي شبكة متصل ببعض المضيفين والمفاتيح وأجهزة التوجيه وتستخدم تقنية افتراضية لجعل النظام مشابها للشبكات الحقيقية. يمكن لهذه الأداة إنشاء شبكة تدعم sdn بسهولة كما يمكن إضافة وظائف جديدة بشكل مرّن إلى الشبكة وإجراء الاختبارات ومن ثم نشر نتائجها بسهولة على أرض الواقع.

ومن مميزات أنها يمكن ببساطة وبسرعة إنشاء مخطط شبكي يدعم تخصيص المستخدم ويقصر دورة التطوير والاختبار حيث هذا يتيح لنا إمكانية تشغيل جميع البرامج التي تعمل على linux باستخدام هذه الأداة كما أنها توفر واجهة تطبيقات بآيثون سهلة الاستخدام.

2-4 السيناريو الأول :

إنشاء شبكة بسيطة مكونة من جهازين h1, h2 موصولين إلى جهاز سويتش s1 والذي بدوره يرتبط مع متحكم c0. ننشئ هذه الشبكة باستخدام الأداة mininet ضمن جلسة اتصال putty عن طريق المنفذ 2223 اعتماداً على localhost وإتمام هذه العملية يتطلب استخدام xmining sever الذي يلعب دور الواجهة التي سنرسم باستخدامها طبولوجيا الشبكة السابقة المراد منها التحقق من الاتصال بين المضيفين باستخدام عملية ال ping الموضحة في الشكل التالي :



الشكل (1-4) طبولوجيا الشبكة المطلوبة

1-2-4 المكونات المستخدمة :

1-1-2-4. المتحكم (controller) :

هو جهاز يفعل الخدمات المراد تقديمها من الشبكة على كل جهاز من أجهزة الشبكة ويديرها ويتحكم بها اتوماتيكيا.

2-1-2-4. المضيف (host):

هو حاسوب او جهاز متصل مع الشبكة يمنح عنوان من فضاء عناوين الشبكة ويطلب الخدمات من المخدم .

3-1-2-4. المبدل (switch):

هو جهاز تشبيك يربط أجزاء الشبكة مع بعضها البعض بواسطة حزم المبدلات لارسال ومعالجة وإيصال البيانات بين الأجهزة ويرتبط بدوره مع المتحكم .

4-1-2-4. أداة ال miniedit :

هي أداة تستخدم لانشاء طبولوجيا الشبكة المطلوبة وهي تمثل الواجهة الرسومية للشبكة .

5-1-2-4. أداة ال putty :

هي تطبيق محاكي طرفي يمكنه إنشاء جلسات اتصال عن بعد باستخدام بروتوكول معين (ssh,telnet) للربط بالسيرفر من أي مكان اعتمادا على المنافذ .

6-1-2-4. xming server:

محاكي طرفي مفتوح المصدر يعمل على أجهزة ويندوز ويسمح لها بعرض برامج لينوكس الرسومية المنفذة على خوادم لينوكس البعيدة ويستخدم بالتزامن مع برنامج العميل putty ssh.

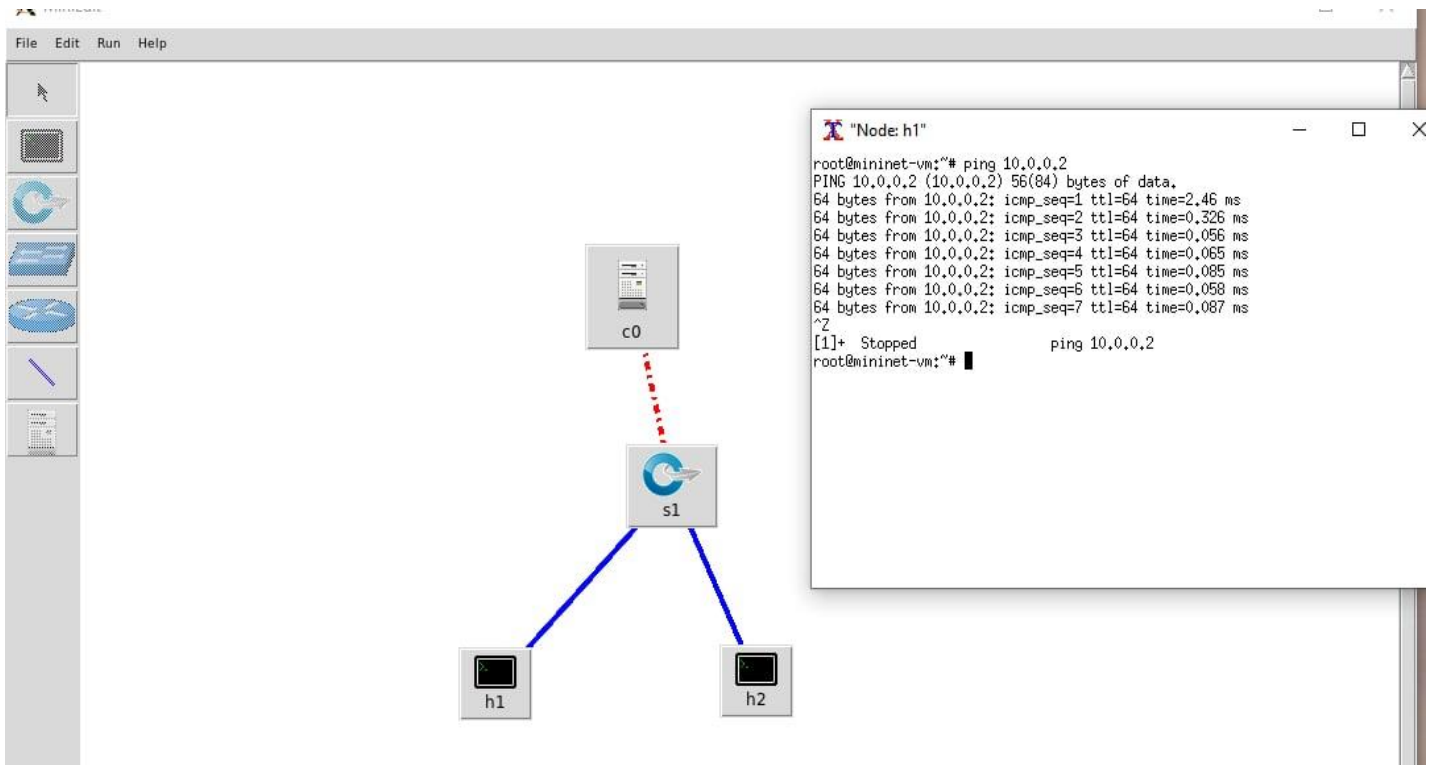
4-2-2 آلية التنفيذ:

أولاً نفتح ال virtual box ونقوم بتشغيل ال mininet والذي يتم اعتماده على اسم مستخدم وكلمة مرور تحمل نفس اسم الأداة (mininet).

ثانياً نقوم بإنشاء جلسة اتصال putty لنتمكن من استخدام الأداة miniedit بالتزامن مع تفعيل ال xming server ولإنشاء هذه الجلسة نفتح ال putty ونختار اسم المستخدم mininet@localhost لربطها مع ال mininet التي أنشأناها على ال virtualbox والمنفذ 2223 ومن ثم ننتقل إلى ال ssh ثم 11x المرتبط بالمتحكم pox ونفعل عملية التوجيه على المنفذ الخاص بالبروتوكول المراد العمل عليه ونبدأ الجلسة..

ثالثاً نبدأ باستدعاء ال miniedit وتظهر لدينا واجهة موضحة الأسفل وعليها سننشئ الطبولوجيا المطلوبة ومن ثم نفعل ال cli الخاص بكل مكون للشبكة ونحدد عنوان الشبكة الذي سيتم إعطاؤه للمضيفين بالتسلسل ومن ثم نقوم بعملية run..

والآن نعود لل putty ونفتح واجهة ال cli الخاصة بكل مضيف باستخدام الأمر xterm ومن ثم نتحقق من الاتصال بعملية ping من أحد المضيفين متبوعاً بعنوان ال ip للمضيف الآخر لنلاحظ الاستجابة المتمثلة بإرسال وتلقي الرزم والتي ستؤكد نجاح الاتصال.



الشكل (2-4) عملية ال ping

3-4 السيناريو الثاني:

قياس إنتاجية الشبكة السابقة باستخدام ال TCP وعرض النتائج بشكل خط بياني.

4-3-1 خطوات العمل:

نتنقل إلى جلسة ال putty ونفتح ال cli الخاصة بكل مضيف حيث سيكون لدينا أحد المضيفين سيرفر وهو الذي سينشئ الاتصال مع المضيف الآخر والذي سيلعب دور الزبون ويتم ذلك باستخدام مجموعة تعليمات سنوضحها تباعاً:

. سوف نحتاج الى استخدام التعليمه iperf والتي هي أداة لاختبار أداء الشبكة تعتمد على / TCP IP و UDP / IP ، وتستخدم للكشف عن استخدام عرض النطاق الترددي للشبكة وجودة الشبكة ، ويمكنها اختبار الحد الأقصى لأداء عرض النطاق الترددي ل TCP و UDP .

يعمل Iperf في وضع C / S ويعمل على الخادم والعميل. يستخدم الخادم بشكل أساسي لمراقبة طلبات الاختبار الواردة ، بينما يستخدم العميل بشكل أساسي لبدء جلسات الاتصال. لذلك ، لاستخدام iperf ، يلزم وجود خادمين ، أحدهما يعمل في وضع الخادم والآخر يعمل في وضع العميل. يمكن استخدام بارامترات مع هذه التعليمه ومنها :

t- :لتحديد وقت التشغيل .

i- :لتحديد معدل الإخراج.

p- :تحديد المنفذ المطلوب.

وتنفيذ هذه التعليمات يتم بشكل متزامن على كلا المضيفين حيث نفتح ال cli الخاص بكل مضيف ونتبع تسلسل العمليات بين المضيفين حيث سيمثل المضيف h2 ال server الذي سيراقب طلبات الاتصال الواردة من ال clinet والممثل بالمضيف h1 الذي سيبدأ جلسات الاتصال المتمثلة ب 15 رساله سيتم مراقبتها وعرضها على المخدم h2 كما يلي:

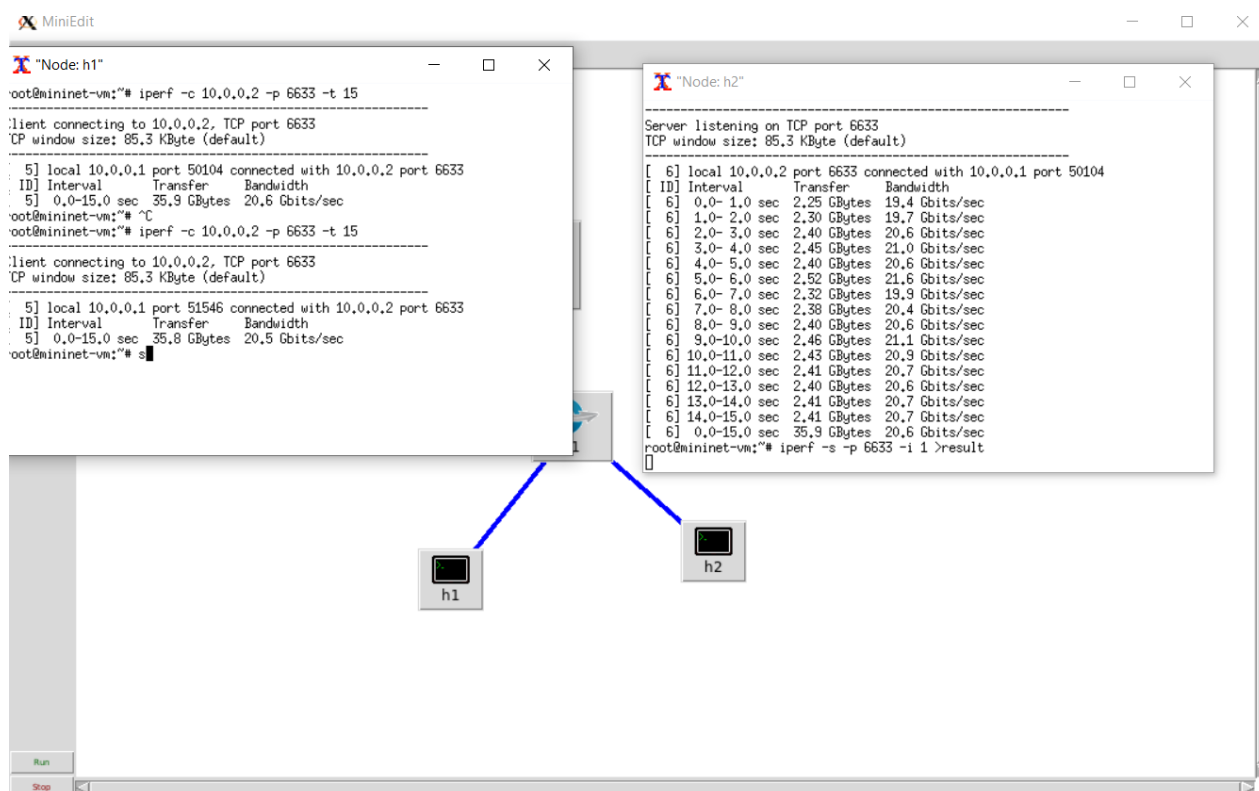
```

"Node: h2"
root@mininet-vm:~# iperf -s -p 6633 -i 1

Server listening on TCP port 6633
TCP window size: 85.3 KByte (default)

```

الشكل (3-4) مراقبة السيرفر (h2) لطلبات المنفذ



الشكل (4-4) طلب الزبون h1 الاتصال مع h2 وحدوث استجابة

والان سنقوم بتخزين الرسائل المتبادلة ضمن ملف ونسميه result يمكن عرض محتواه باستخدام التعليمة more result والان سنوجه محتويات هذا الملف إلى ملف اخر لاستخدامه من قبل الزبون أثناء عرض الخط البياني المعبر عن النتائج ونسميه new_result .

```

MiniEdit
File Edit Run Help

"Node: h2"
-----
Server listening on TCP port 6633
TCP window size: 85,3 KByte (default)
-----
[ 6] local 10.0.0.2 port 6633 connected with 10.0.0.1 port 44556
[ ID] Interval      Transfer    Bandwidth
[ 6] 0.0- 1.0 sec   630 MBytes  5,28 Gbits/sec
[ 6] 1.0- 2.0 sec   1,34 GBytes 11,5 Gbits/sec
[ 6] 2.0- 3.0 sec   1,39 GBytes 12,0 Gbits/sec
[ 6] 3.0- 4.0 sec   1,43 GBytes 12,3 Gbits/sec
[ 6] 4.0- 5.0 sec   1,41 GBytes 12,1 Gbits/sec
[ 6] 5.0- 6.0 sec   1,39 GBytes 11,9 Gbits/sec
[ 6] 6.0- 7.0 sec   1,43 GBytes 12,3 Gbits/sec
[ 6] 7.0- 8.0 sec   1,41 GBytes 12,1 Gbits/sec
[ 6] 8.0- 9.0 sec   1,41 GBytes 12,1 Gbits/sec
[ 6] 9.0-10.0 sec   1,42 GBytes 12,2 Gbits/sec
[ 6] 10.0-11.0 sec  1,36 GBytes 11,7 Gbits/sec
[ 6] 11.0-12.0 sec  1,31 GBytes 11,3 Gbits/sec
[ 6] 12.0-13.0 sec  1,42 GBytes 12,2 Gbits/sec
[ 6] 13.0-14.0 sec  1,41 GBytes 12,1 Gbits/sec
[ 6] 14.0-15.0 sec  1,42 GBytes 12,2 Gbits/sec
[ 6] 0.0-15.0 sec  20,2 GBytes 11,6 Gbits/sec
^Croot@mininet-vm:~# iperf -s -p 6633 -i 1 > result

"Node: h1"
-----
root@mininet-vm:~# iperf -c 10.0.0.2 -p 6633 -t 15
-----
Client connecting to 10.0.0.2, TCP port 6633
TCP window size: 85,3 KByte (default)
-----
[ 5] local 10.0.0.1 port 44556 connected with 10.0.0.2 port 6633
[ ID] Interval      Transfer    Bandwidth
[ 5] 0.0-15.0 sec  20,2 GBytes 11,6 Gbits/sec
root@mininet-vm:~# ^C
root@mininet-vm:~# ^C
root@mininet-vm:~# ^C
root@mininet-vm:~# iperf -c 10.0.0.2 -p 6633 -t 15
-----
Client connecting to 10.0.0.2, TCP port 6633

```

الشكل (4-5) تخزين الرسائل ضمن الملف result

```

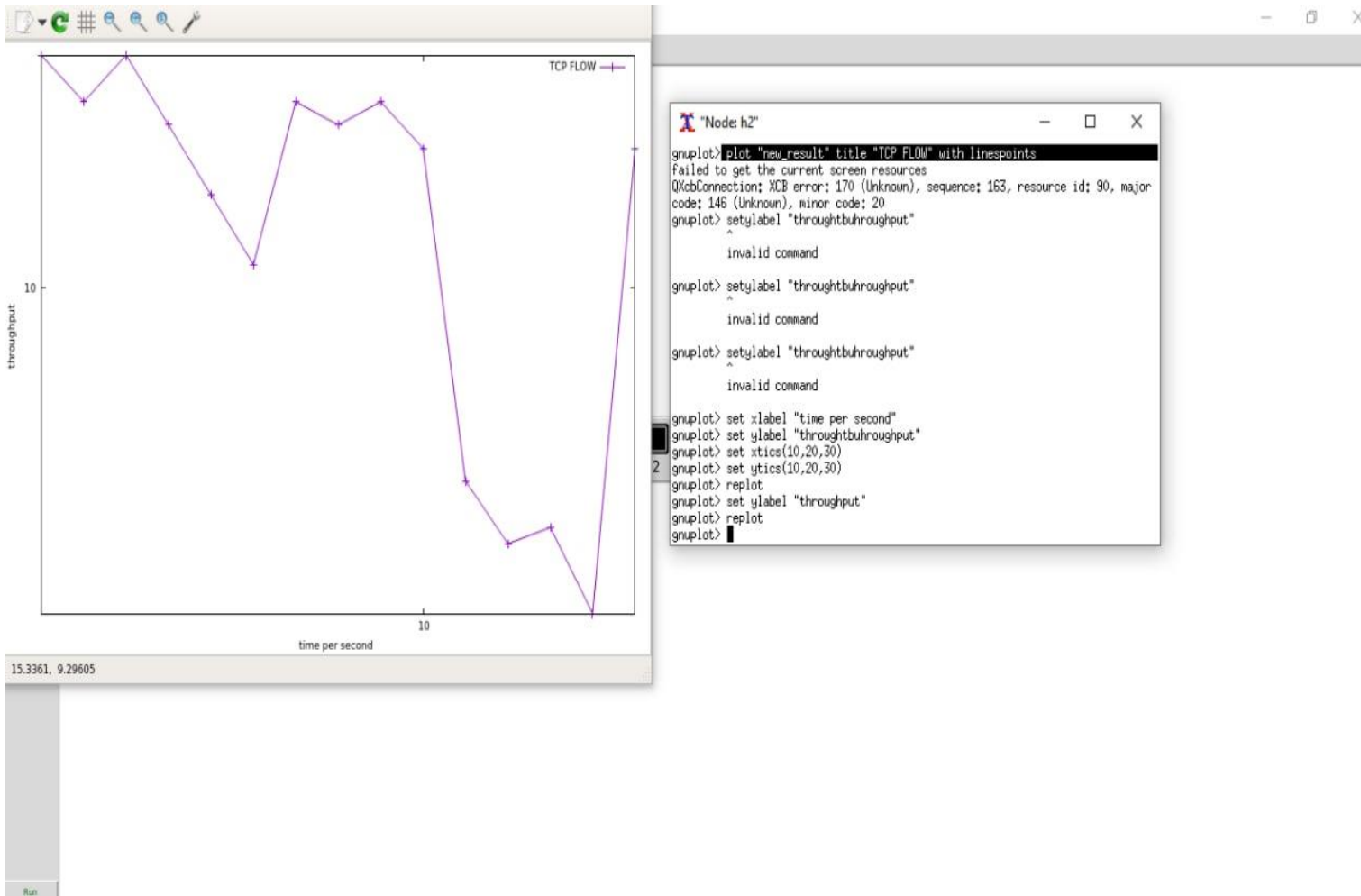
MiniEdit
File Edit Run Help

"Node: h1"
-----
root@mininet-vm:~# iperf -c 10.0.0.2 -p 6633 -t 15
-----
Client connecting to 10.0.0.2, TCP port 6633
TCP window size: 85,3 KByte (default)
-----
[ 5] local 10.0.0.1 port 44556 connected with 10.0.0.2 port 6633
[ ID] Interval      Transfer    Bandwidth
[ 5] 0.0-15.0 sec  20,2 GBytes 11,6 Gbits/sec
root@mininet-vm:~# ^C
root@mininet-vm:~# ^C
root@mininet-vm:~# ^C
root@mininet-vm:~# iperf -c 10.0.0.2 -p 6633 -t 15
-----
Client connecting to 10.0.0.2, TCP port 6633

```

الشكل (4-6) توجيه محتوى الملف result إلى ملف new_result

والآن ننتقل إلى المرحلة النهائية من السيناريو الثاني وسننتقل إلى وضع ال gnuplot والذي سيتم رسم الخط البياني بالمواصفات المطلوبة بالاعتماد عليه كما يلي:



الشكل (4-7) الخط البياني للإنتاجية باستخدام TCP

الخاتمة:

تقنية ال SDN تم اختراعها لتسهيل عملية التحكم في الشبكة بشكل عام من ناحية ادارة و تحكم و في الشبكة ، و العنصر الاساسي الذي تم الاعتماد عليه في بناء تقنية ال SDN هو بروتوكول ال أوبن فلو الذي قمنا بتحليل أدائه و قد تم حل الكثير من المشاكل التي كان مهندس الشبكة يعاني منها و عندما تم اختراع هذه التقنية تم حل جميع المشاكل واصبح ادارة الشبكة اسهل بكثير حيث أن مهندس الشبكة يستطيع التحكم في الشبكة من خلال البرامج.

ويجب على مهندس الشبكة أن يكون على معرفة في لغة البرمجة مثل جافا أو بايثون أو روبي ليستطيع عمل برامج للتحكم في الشبكة و يجب أن يكون على معرفة ممتاز في عالم الشبكة ليستطيع عمل هذه البرامج ، وعلى مهندس الشبكة أن يكون بمستوى المحترفين على الاقل ليستطيع العمل في هذا الموضوع ويشترط أن يكون ايضاً على معرفة و دراسة في

خاصية و تقنية التطبيقات الوهمية virtualization

يساعد هذا المشروع في فهم مكونات التقنية حيث وجدنا أنها تقنية حديثة حيث إن شبكات اليوم ببنيتها التحتية القديمة غير جاهزة للعمل بهذه التقنية وبالتالي سيحتاج هذا الأمر للكثير من الجهود المتضافرة من الشركات المصنعة والباحثين والمهندسين والشركات المستفيدة حيث يستغرق هذا عدة سنوات حيث بدأت تقنية SDN بالظهور بالفعل في ارض الواقع لكن فقط على مستوى مراكز البيانات Data Centers وبعض طبقات تصميم البنية التحتية للشبكات العملاقة.

الاستنتاجات والأعمال المستقبلية:

تناولنا في هذا البحث محاكاة شبكة sdn حيث قمنا بتحليل ونمذجة شبكات SDN من خلال تطبيق طوبولوجيات SDN باستخدام أداة Mininet و قمنا بإرسال باكيتات عبر الشبكة ودراسة التأخير وسرعة النقل ضمن الشبكة سيتم العمل في المستقبل على تقييم أداء أكثر من متحكم وإجراء مقارنة بينها.

1. Ahmed Sonba, Hassan Abdalkreim "Performance Comparison Of the state of the art Openflow Controllers".
2. Nick Feamster, Jennifer Rexford, Ellen Zegura, "The Road to SDN: An Intellectual History of Programmable Networks". ACM SIGCOMM, Volume 44 Issue 2, 2014, 87-98
3. F. d. O. Silva, J. H. d. S. Pereira, P. F. Rosa, and S. T. Kofuji, "Enabling Future Internet Architecture Research and Experimentation by Using Software Defined Networking," in European Workshop on Software Defined Networking (EWSDN), pages 73-78, Oct. 2012
4. H. Kim and N. Feamster. "Improving Network Management with Software Defined Networking," IEEE Communication Magazine, vol. 51, pages 114-119, Feb. 2013
5. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, 1299J. Rexford, S. Shenker, J. Turner, OpenFlow: enabling innovation in 1300 campus networks, ACM SIGCOMM CCR 38 (2) (2008) 69–74.
6. SOUAD, F.; MOUGHIT,M. ; IDBOUFKER, N. OpenFlow Controllers Performance Evaluation. International Journal of Emerging Research in Management &Technology 2016
7. Fancy,C. ; Pushpalatha,M. Performance Evaluation of SDN controllers POX and Floodlight in Mininet Emulation Environment. IEEE, 2017.

