مجلة جامعة طرطوس للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (٧) العدد(٦) ٢٠٢٣
Tartous University Journal for Research and Scientific Studies - engineering Sciences Series Vol. (7) No. (6) 2023

تقييم أداء الشبكات المعرفة برمجيا (SDN) اعتماداً على نوع المتحكم

أ.د.م فادي غصنه*

م. ثراء حسن**

(تاريخ الإيداع ٢٠٢٣/٦/١٨ . قُبِل للنشر في ٢٠٢٣/٨/١٣)

□ ملخّص□

تهدف الشبكات المعرفة برمجيا إلى التغلب على العقبات التي تواجه الشبكات التقليدية، وتعد عملية فصل اتخاذ قرارات التوجيه عن عملية توجيه البيانات هي جوهر تقنية الشبكات المعرفة بالبرمجيات SDN. تعمل هذه الفكرة على تحسين عمل الشبكة بعدة طرق، مثل الاستخدام الفعال للموارد وإدارة أفضل للشبكة بالإضافة إلى تقليل التكلفة. ومن المكونات الأساسية لهذه التقنية هو المتحكم المسؤول عن التحليل والمراقبة لحركة البيانات. وبالتالي هناك حاجة كبيرة لوحدة تحكم فعالة من أجل تحقيق أداء أفضل في الشبكات ومراكز البيانات، بالإضافة إلى توفير توجيه فعال لحركة المرور، والذي بدوره يؤدي إلى تحسين استخدام الموارد.

قدم هذا البحث دراسة عن تأثير المتحكمين RYU و POXعلى أداء الشبكات المعرفة برمجيا، حيث تم تقييم أداء الشبكات من خلال دراسة البارامترات (عرض الحزمة، الإنتاجية، زمن الإرسال والاستقبال) لطوبولوجيا شبكات ال SDN (الأصغرية، المفردة، الخطية، الشجرية).

الكلمات المفتاحية: SDN، حركة المرور، متحكم، POX،RYU .

*أستاذ في قسم هندسة تكنولوجيا الاتصالات – كلية هندسة تكنولوجيا المعلومات والاتصالات – جامعة طرطوس – سوريا
** طالبة ماجستير – قسم هندسة تكنولوجيا الاتصالات – كلية هندسة تكنولوجيا المعلومات والاتصالات – جامعة طرطوس - سوريا

مجلة جامعة طرطوس للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (٧) العدد(٦) ٢٠٢٣
Tartous University Journal for Research and Scientific Studies - engineering Sciences Series Vol. (7) No. (6) 2023

Performance evaluation of Software-Defined networks Based on Controller Type

ph.Dr.Eng. fadi Ghosna*
Eng. Tharaa Hasan**

(Received 18/6/2023 . Accepted 13/8/2023)

□ **ABSTRACT**

Software-defined network (SDN) is a new approach that overcomes the obstacles faced by conventional network. Decoupling the decision-making process from the data forwarding process is the heart of software-defined networks technology. This idea improves the network in many ways, such as efficient utilization of resources, better management of the network, and reduced cost. One of the most important components of this technology is the controller, which is responsible for the analysis and monitoring of data traffic. There is a great need for a high-performance controller in networking industries, data centers, Therefore, it is crucial to investigate the performance of an open-source controller to provide efficient traffic routing, leading to improved utilization of resources for the enhanced performance metrics of the network.

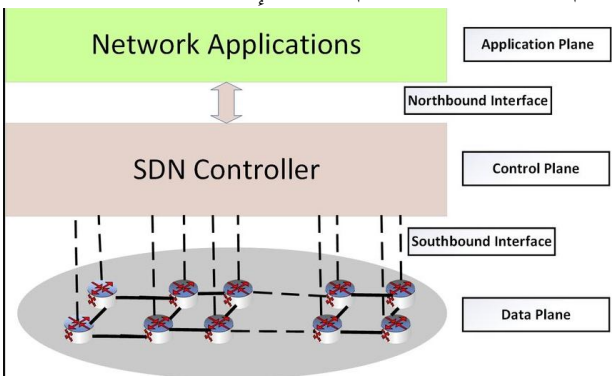
In this work, a comparative study has between considered. This study comparison between RYU controller and POX controller effect an SDN performance. The performance has been evaluated using the parameters: bandwidth, throughput, round-trip time, of the SDN topologies (minimal, single, linear, tree).

Keywords: SDN, Traffic Network, SDN Controller, RYU, POX.

*Professor, communication Technology Engineering Department, Information and communication Technology Engineering, Tartous University, Syria.
** Master Student – communication Technology Engineering Department, Information and communication Technology Engineering, Tartous University, Syria

١ – مقدمة:

تعد الشبكات المعرفة بالبرمجيات تقنية ديناميكية قابلة للبرمجة وذات هيكلية منظمة بشكل جيد[1]. حيث قدمت العديد من الميزات وساهمت في حل العديد من المشاكل التي عانت منها الشبكات التقليدية. تعتمد الشبكات المعرفة بالبرمجيات على فصل عمليات توجيه البيانات عن عملية التحكم واتخاذ قرارات التوجيه لجعل التحكم بالشبكة مركزياً [2]. وبالتالي تقليل الأخطاء التي من الممكن أن تتعرض لها الشبكات. تتألف الشبكات المعرفة بالبرمجيات من ثلاث طبقات وهي طبقة التطبيقات (Application layer) وطبقة التحكم(control layer) والطبقة الفيزيائية (physical layer)، تحتوي طبقة التطبيقات على التطبيقات المصممة لتلبي رغبات المستخدمين، بينما تصل طبقة التحكم بين طبقة التطبيقات والطبقة الفيزيائية وتحتوي على المتحكم الذي يعد 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، وبالنسبة للطبقة الفيزيائية فهي تتألف من أجهزة التبديل مثل المبدلات والموجهات ولها مهمتين الأولى تجميع حالة الشبكة وإرسالها إلى المتحكم والثانية معالجة الرزم المرتكزة على قوانين منصبة من قبل المتحكم. يتم الاتصال بين مستوى التحكم ومستوى البيانات بمساعدة واجهة التخاطب الجنوبية (API)، بينما الاتصال بين مستوى التحكم ومستوى الإدارة يتم باستخدام الواجهة التخاطب الشمالية API كما هو موضح في الشكل (1). تعتبر المتحكمات هي العقل المدبر وجوهر الشبكات المعرفة بالبرمجيات، حيث يوجد العديد من المتحكمات مثل beacon، maestro، NOX، POX، RYU، openDaylight ODL، ONOS، floodlight وبالتالي يعمل الاختيار المناسب للمتحكم على تحسين استخدام الموارد وإدارة الشبكة بشكل أفضل [3].



الشكل (١): واجهات التخاطب في الشبكات المعرفة برمجيا

❖سنستعرض بعض الدراسات التي أجريت في مجال الشبكات المعرفة برمجيا:

▪**الدراسة الأولى:** اقتراح نهج جديد لشبكات SDN من أجل مراقبة حركة المرور وتحليها بناء على مجموعة من البيانات، حيث حدد الباحثون استهلاك عرض الحزمة لموارد الشبكة في بيئة SDN، وتم اقتراح المزيد من التحسينات [4].

▪**الدراسة الثانية:** مقارنة بين نوعين من المتحكمات مفتوحة المصدر ODL وONOS، وكانت النتيجة أن أداء المتحكم ODL أفضل بكثير من المتحكم ONOS بالاستناد إلى بارامترات الأداء مثل الإنتاجية وعرض الحزمة [5].

▪**الدراسة الثالثة:** مقارنة أداء المتحكمات NOX وRYU و Floodlightو POXحيث تم إجراء المقارنة بناء على سعة الرزم باستخدام (D-ITG) وكانت النتيجة أن المتحكم Floodlight قدم أداء أفضل من حيث الإنتاجية والتأخير مقارنة بالمتحكمات الأخرى [6].

▪**الدراسة الرابعة:** مقارنة أداء المتحكمات Floodlight وpox، حيث تم تقديم طوبولوجيا SDN مختلفة لكل من المتحكمات ومقارنة الأداء بناء على الإنتاجية وRTT، وكانت النتيجة أن المتحكم Floodlight لدية أداء أفضل من المتحكم POX [7].

٢-أهمية البحث وأهدافه:

تعد عملية تحليل حركة المرور من المواضيع المهمة في مجال تطبيقات الشبكات المعرفة بالبرمجيات SDN وذلك نتيجة التطور الكبير في مجال التكنولوجيا. وهذا بدوره يتطلب الاختيار المناسب للمتحكم لأنه أساس عمل الشبكات المعرفة بالبرمجيات، وأيضا المسؤول عن تحليل ومراقبة حركة البيانات وإدارة الشبكة بشكل أفضل.

يهدف هذا البحث لتقييم أداء الشبكات المعرفة بالبرمجيات من خلال دراسة تأثير المتحكم RYU والمتحكم POX على الشبكات وتقييم الأداء بدراسة البارامترات (عرض الحزمة Bandwidth، الإنتاجية Throughput، زمن الذهاب والإياب Round Trip Time).

٣-طرائق البحث ومواده:

لتحقيق هدف البحث تم اتباع المنهجية الآتية:

❖**دراسة النظرية:** تضمنت التعريف بالمتحكم RYU والمتحكم POX، وأنواع بنى الشبكات المعرفة برمجيا.

❖**الدراسة العملية:** تم استخدام المحاكي Mininet لبناء سيناريوهات لأنواع بنية الشبكات المعرفة بالبرمجيات (**minimal**, **single**, **linear**, **tree**) باستخدام المتحكم RYU وPOXو تنفيذ السيناريوهات من أجل تحليل الأداء من حيث الإنتاجية وعرض الحزمة وزمن الذهاب والإياب.

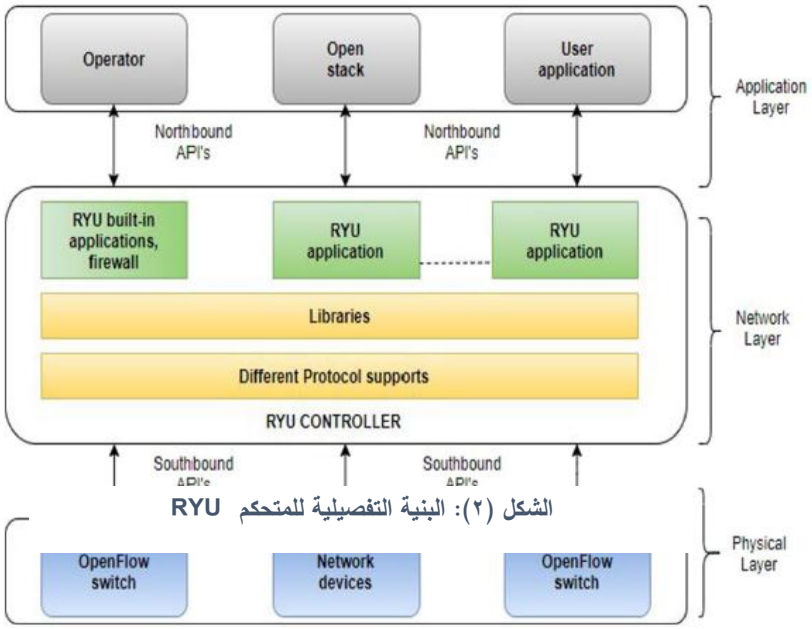
٣-١المتحكم RYU

هو متحكم مفتوح المصدر مكتوب لغة بايثون، يقدم مكونات معرفة بشكل جيد حيث تمكن المطورين من كتابة تطبيقات عديدة للشبكة. يدعم المتحكم معظم إصدارات بروتوكول OpenFlow من 1.0 حتى 1.5. يتم تصنيف بنية المتحكم RYU SDN إلى ثلاث مستويات مختلفة: طبقة التطبيقات وطبقة الشبكة والطبقة الفيزيائية [8].

الطبقة الدنيا هي الطبقة الفيزيائية، والتي تتكون من أجهزة فيزيائية وافتراضية مختلفة متصلة عبر الإنترنت للتواصل مع بعض. تُعرف هذه الطبقة أيضًا باسم طبقة البنية التحتية لأنها هي الطبقة الأساسية والتي تتكون من أجهزة مختلفة موضوعة على نفس المستوى.

الطبقة الوسطى هي طبقة الشبكة التي تُعرف أيضًا باسم طبقة التحكم لأن في هذه الطبقة يتم التحكم في تدفق حركة البيانات من عقدة إلى عقدة أخرى للحفاظ على الاستقرار في الشبكة بدون وجود حمل. يتم توفير الواجهة بين الطبقة الفيزيائية وطبقة الشبكة بمساعدة واجهات برمجة التطبيقات المعروفة باسم الواجهة الجنوبية.

الطبقة العليا هي طبقة التطبيقات التي تتكون من تطبيقات المستخدم النهائي مثل الشبكة والأعمال المنطقية. يتم توفير الواجهة بين طبقة الشبكة وطبقة التطبيق بمساعدة واجهات برمجة التطبيقات المعروفة باسم الواجهة الشمالية [9]. يبين الشكل (٢) البنية التفصيلية للمتحكم RYU.



الشكل (٢): البنية التفصيلية للمتحكم RYU

٢-٣المتحكم POX:

يعد المتحكم pox من أشهر المتحكمات المستخدمة ضمن الشبكات المعرفة بالبرمجيات، وهو عبارة عن وحدة تحكم مفتوحة المصدر أيضا، تعتمد على لغة python.، حيث يدعم هذا المتحكم أيضا بروتوكول OpenFlow. وتتألف بنيت من العديد من المكونات مثل واجهة التحكم: التي تستخدم للتفاعل بين المتحكم والشبكة المستخدمة، قارئ الحزم: يقوم هذا المكون بقراءة الحزم الواردة من المبدل وإعادة توجيها إلى المكان المناسب، وحدة التخزين: تستخدم لتخزين البيانات اللازمة من أجل عملية اتخاذ القرارات، وحدة تحكم المبدل: وهي وحدة تستخدم لإرسال الأوامر من المبدل إلى الموجه، وحدة المراقبة: تساعد في مراقبة حالة الشبكة وإصدار التنبيهات عند الحاجة.

3-3 أنواع بنى الشبكات المعرفة برمجيا

❖البنية الأصغرية (Minimal Topology):

تعد هذه البنية الأصغر ضمن الشبكات المعرفة برمجيا حيث تتألف من مبدل واحد s1 ومتصل معه مضيفين h1 و h2.

❖البنية المفردة (single topology):

تعد هذه البنية من البنى البسيطة حيث تتألف من مبدل واحد فقط s1 وعدد من المضيفين N متصل بالمبدل، يرمز لها ب single,x، حيث تدل x على عدد المضيفين المتصلين بالمبدل الوحيد.

❖البنية الخطية (Linear Topology):

تتألف هذه البنية من N مبدل متصل معه N مضيف، يرمز لها linear,x حيث تدل x على عدد المبدلات والمضيفين.

❖البنية الشجرية (Tree Topology):

تعد هذه البنية متعددة الطبقات حيث يرمز لها tree,x,y، يدل x على عمق الشبكة بينما y على عدد أبناء الشبكة.

٣-٢ الدراسة العملية:

❖أداة البحث:

من أجل تنفيذ البحث تم استخدام المحاكي Mininet [10]، وهو محاكي ينشئ شبكة من المضيفات الظاهرية والمبدلات والمتحكمات والروابط. تستطيع مضيفات Mininet تشغيل برامج نظام Linux القياسية، كما تدعم البروتوكول OpenFlow لتحقيق توجيه مخصص للرزم عالي المرونة. حيث تم إنشاء ثلاثة أنواع من بنى SDN وتم استخدام المتحكم RYU والمتحكم POX وتحليل أداء الشبكات من خلال دراسة البارامترات (عرض الحزمة، الإنتاجية، زمن الذهاب والإياب) حيث تم استخدام الأداة Iperf من أجل قياس عرض الحزمة للشبكة التي تم إنشائها، والأداة iperf3 من أجل قياس الإنتاجية. مواصفات الجهاز الذي تم استخدامه عند القيام بالتجارب هي: جهاز بنظام تشغيل Ubuntu وبمعالج intel® Core™i3-6006U CPU @ 2.00GHz 2.00GHz وذاكرة وصول عشوائية ٤ غيغابايت.

٤-النتائج والمناقشة:

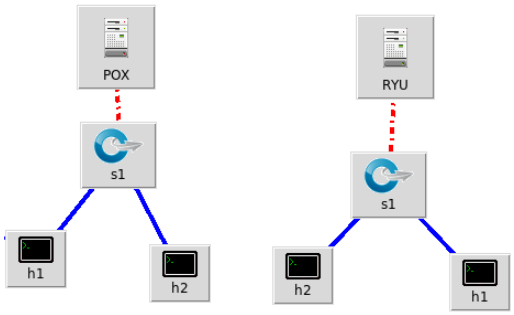
تم تنفيذ عدة سيناريوهات لبنية الشبكات وفي كل سيناريو تم قياس ما يلي:

- الإنتاجية من خلال الأداة iperf3 بين المضيفين حيث يستخدم المضيف الأول كمخدم والثاني كزبون وتم تنفيذ الأداة لمدة ١٠ ثواني. من خلال التعليمة Iperf3 –s –p 5566 –i 1 هنا يصبح المضيف كمخدم والمضيف الثاني كزبون من خلال التعليمة iperf3 –c 10.0.0.2 –p 5566 –t 10 . وعند الانتهاء من القياس أخذ القيمة المتوسطة للإنتاجية.

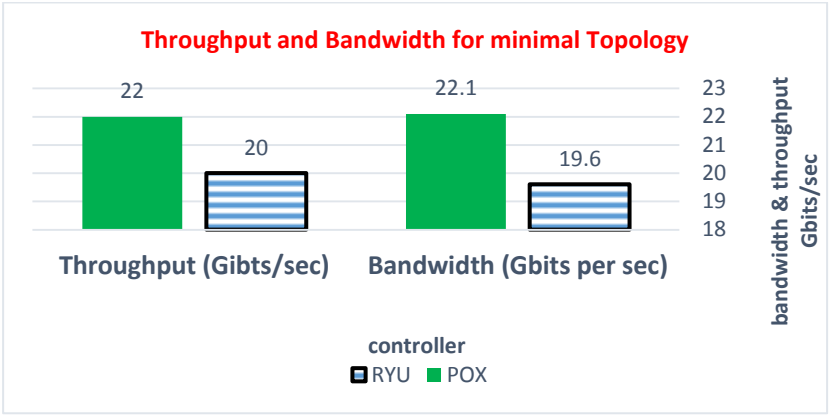
حيث s: للدلالة على أن المضيف سيرفر، p: للدلالة على رقم المنفذ الخاص بالخدمة، i: لتحديد فترة مراقبة النتيجة كل زمن معين (١ ثانية)، c: للدلالة على أن المضيف زبون، ١٠.٠.٠.٢ تحديد العنوان الذي سيتم الاستقبال منه، t: ضبط زمن الإرسال كل ١٠ ثانية.

- عرض الحزمة تم القياس من خلال الأداة Iperf عند بدء المحاكاة بين المضيفين مثال على ذلك Iperf h1 h2. وعند الانتهاء من قياس عرض الحزمة للمضيفين في الشبكة تم أخذ القيمة المتوسطة لعرض الحزمة.
- زمن الذهاب والإياب تم قياسه من خلال إرسال رزم ICMP من أحد المضيفين واستقبال الرد لها. H1 ping -c3 h2

السيناريو الأول: تم إنشاء البنية الأصغرية أو الافتراضية ضمن برنامج Mininet للشبكات المعرفة بالبرمجيات باستخدام المتحكم RYU وPOX. تتألف هذه البنية من مبدل واحد ومضيفين، يبين الشكل(٣) البيئة الافتراضية للشبكات المعرفة بالبرمجيات، حيث تم قياس عرض الحزمة عند بدء المحاكاة، ثم قياس الإنتاجية بين المضيفين وكانت النتيجة كما في الشكل (٤) من أجل عرض الحزمة والإنتاجية، وبالنسبة ل زمن الذهاب والإياب قمنا بإرسال ثلاثة رزم h1 إلى h2 في البنية باستخدام RYU وكذلك بالنسبة للبنية باستخدام POX، يبين الشكل (٥) زمن الذهاب والإياب بين h1 و h2 بالبنية المستخدمة المتحكم RYU، يوضح الشكل (٦) زمن للذهاب والإياب في البنيتين، أظهرت النتائج أن المتحكم pox لديه أداء أفضل من ناحية عرض الحزمة والإنتاجية بينما المتحكم RYU كان أفضل من ناحية زمن الذهاب والإياب.

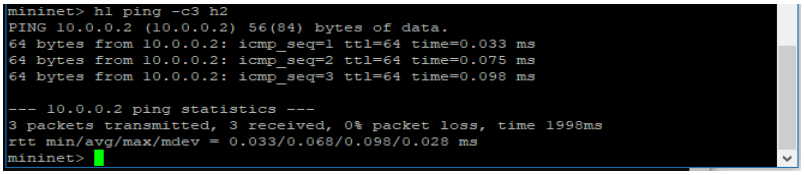


الشكل (٣) البنية الافتراضية باستخدام المتحكمين RYU وPOX

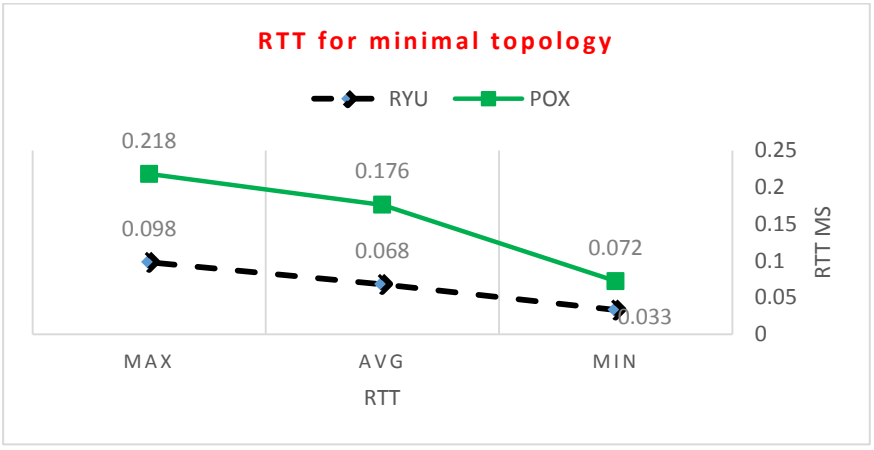


الشكل (٤): يمثل عرض الحزمة والإنتاجية لبنية SDN الافتراضية باستخدام المتحكم RYU وPOX

يمكن ملاحظة أن عرض الحزمة والإنتاجية في البنية المستخدمة المتحكم POX كان أفضل ويعود السبب إلى أن بنية المتحكم POX أفضل من خلال معالجة العمليات من المتحكم RYU.



الشكل (٥) زمن الذهاب والإياب بين المضيفين h1 وh٢ بالبنية المستخدمة المتحكم RYU.

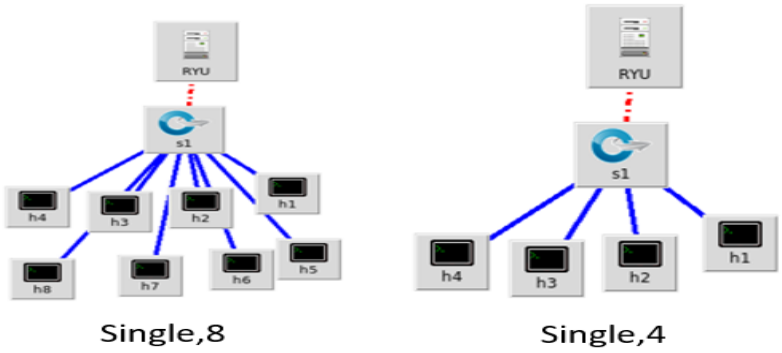


الشكل (٦): زمن الذهاب والإياب لبنية SDN الأصغرية باستخدام المتحكم RYU وPOX

نلاحظ أن زمن الذهاب

والإياب للرزم في البنية المستخدمة المتحكم RYUأفضل من POX ويعود السبب إلى أن المتحكم POX يأخذ فترة أطول لتحديد مسارات الشبكة الموجودة.

❖ **السيناريو الثاني:** تم إنشاء البنية **single,x** للشبكات المعرفة بالبرمجيات باستخدام المتحكم RYU و POXحيث تتألف هذه البنية من **مبدل واحد وx مضيف**، أجريت المقارنة للشبكات باستخدام المتحكمين من أجل single,4 و single,8 و single,25 و single,100 و single,500 و single,1000 يوضح الشكل (٧) بنية single,4 و single,8 باستخدام المتحكم RYU، وتختلف البنى الأخرى لل single فقط بازدياد عدد المضيفين من ٢٥ إلى ١٠٠٠.



الشكل (٧) بنية single,4 و بنية single,8

يمكن أيضا بناء البنية عن طريق كتابة السطر التالي،

Sudo mn --controller=remote,ip=127.0.0.1—mac --switch=ovsk,protocols =OpenFlow13 --topo=single,1000

--controller: تحدد نوع المتحكم وهنا تم اختياره عن بعد وإعطاءه العنوان ip=127.0.0.1

--mac: توضع من أجل إنشاء تخطيط بين عناوين MACو IP وفق عناوين Ip

--switch : تحدد نوع السويتش المستخدم، وتم اختيار البروتوكول OpenFlow.

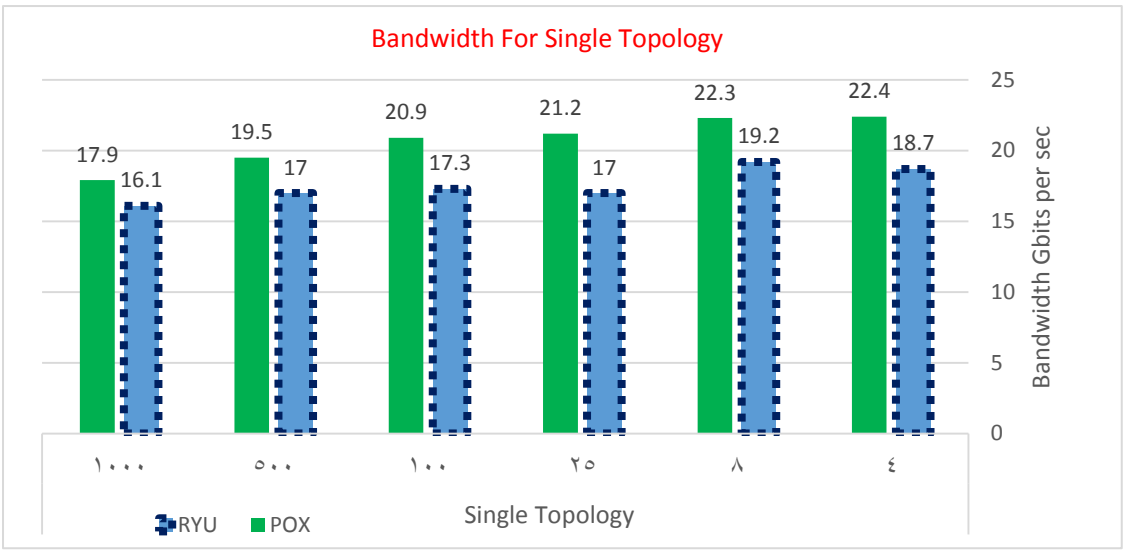
--topo: تحدد نوع البنية، تم اختيار single بمبدل واحد و 1000 مضيف.

يوضح الشكل (٨) بناء البنية single,1000 :

```
mininet@mininet-vm:~$ sudo mn --controller=remote,ip=127.0.0.1 --mac --switch=ov
sk,protocols=OpenFlow13 --topo=single,1000
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Unable to contact the remote controller at 127.0.0.1:6633
Setting remote controller to 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h
23 h24 h25 h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h
43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h
63 h64 h65 h66 h67 h68 h69 h70 h71 h72 h73 h74 h75 h76 h77 h78 h79 h80 h81 h82 h
83 h84 h85 h86 h87 h88 h89 h90 h91 h92 h93 h94 h95 h96 h97 h98 h99 h100 h101 h10
2 h103 h104 h105 h106 h107 h108 h109 h110 h111 h112 h113 h114 h115 h116 h117 h11
8 h119 h120 h121 h122 h123 h124 h125 h126 h127 h128 h129 h130 h131 h132 h133 h13
4 h135 h136 h137 h138 h139 h140 h141 h142 h143 h144 h145 h146 h147 h148 h149 h15
0 h151 h152 h153 h154 h155 h156 h157 h158 h159 h160 h161 h162 h163 h164 h165 h16
6 h167 h168 h169 h170 h171 h172 h173 h174 h175 h176 h177 h178 h179 h180 h181 h18
2 h183 h184 h185 h186 h187 h188 h189 h190 h191 h192 h193 h194 h195 h196 h197 h19
8 h199 h200 h201 h202 h203 h204 h205 h206 h207 h208 h209 h210 h211 h212 h213 h21
4 h215 h216 h217 h218 h219 h220 h221 h222 h223 h224 h225 h226 h227 h228 h229 h23
0 h231 h232 h233 h234 h235 h236 h237 h238 h239 h240 h241 h242 h243 h244 h245 h24
6 h247 h248 h249 h250 h251 h252 h253 h254 h255 h256 h257 h258 h259 h260 h261 h26
```

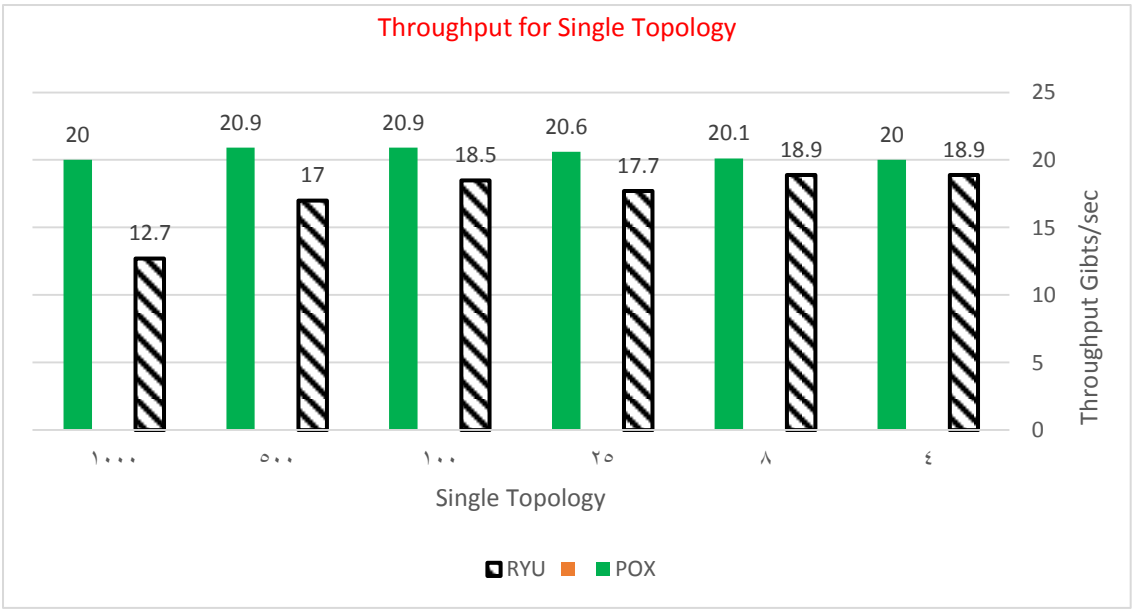
الشكل (٨) بناء بنية single,1000 في Mininet

يوضح الشكل (٩) عرض الحزمة للبنى المختلفة، والشكل (١٠) الإنتاجية للبنى المختلفة والشكل (١١) زمن الذهاب والإياب.



الشكل (٩): عرض الحزمة للبنية Single باستخدام المتحكم RYU وpox

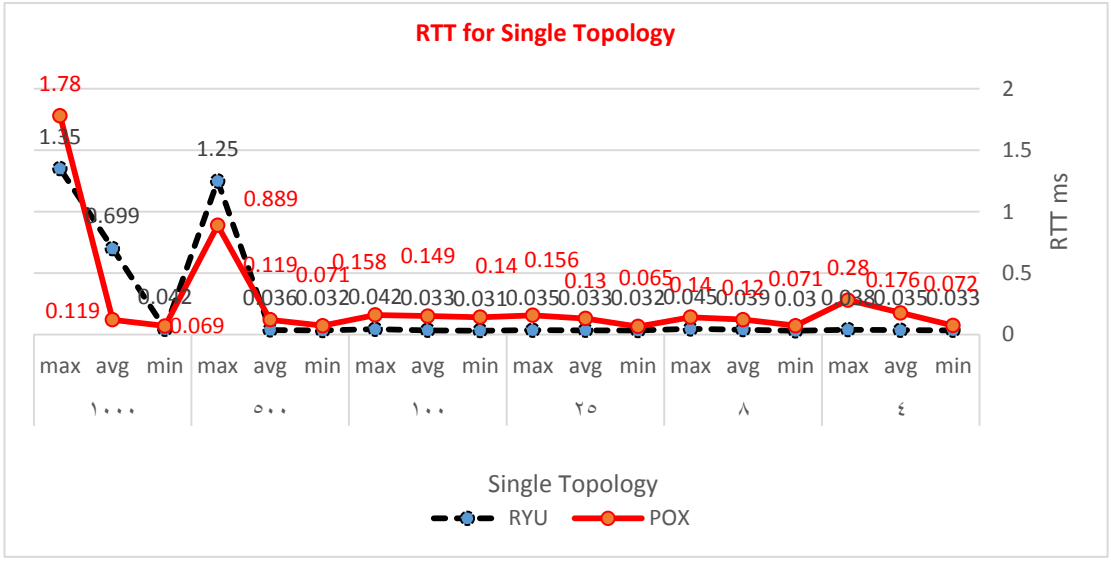
يمكن ملاحظة من خلال الشكل (٩) ان عرض الحزمة عند استخدام المتحكم RYU بقي ثابتا تقريبا عند ازياد عدد المضيفين عن ٢٥ مضيف، أي أنه يحافظ على سوية الأداء نفسها وزيادة عدد المضيفين لا يؤثر على أداء المتحكم. أما بالنسبة المتحكم POX يمكن ملاحظة أن عرض الحزمة تناقص بشكل مستمر مع ازياد عدد المضيفين أي أن قدرته على معالجة البيانات تنخفض مع ازياد عدد المضيفين، وبالوقت نفسه ومع انخفاض عرض الحزمة للبنية المستخدمة المتحكم POX إلا أنه يقدم عرض حزمة أفضل من RYU، والسبب كما ذكرنا سابقا قدرته العالية على معالجة كمية أكبر من البيانات مقارنةً مع المتحكم RYU.



الشكل (١٠): الإنتاجية للبنية Single باستخدام المتحكم RYU والمتحكم POX

نلاحظ من الشكل (١٠) أن الإنتاجية عند استخدام المتحكم RYU حافظت على نفس القيم تقريبا، ولكن عند ازياد عدد المضيفين من ٥٠٠ ل 1000 انخفضت بشكل كبير نتيجة ازدياد عدد العمليات التي تحتاج للمعالجة وهذا بدوره يؤدي

لاحتماليه فقد بعض الرزم لذلك انخفضت الإنتاجية، أما بالنسبة للمتحكم POX كان لديه أداء أفضل من RYU وبقي محافظ على الإنتاجية مع ازياد عدد المضيفين ويعود ذلك لامتلاكه آلية معالجة للرزم أفضل من المتحكم RYU.



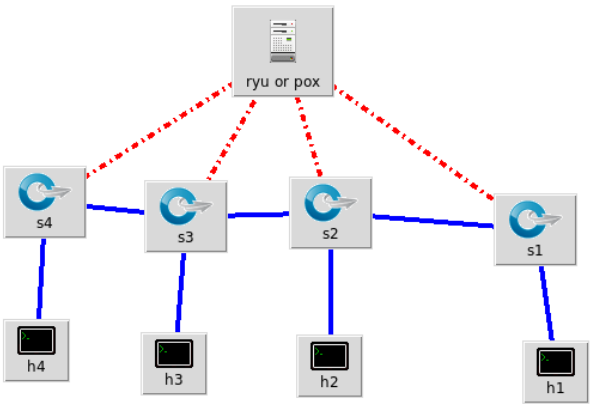
الشكل (١١): زمن الذهاب والإياب للبنية Single باستخدام المتحكم RYU وpox

نلاحظ من الشكل (١١) أن المتحكم RYU و POXلديهم زمن ذهب وإياب منخفض ولكن عند ازياد عدد المضيفين أكثر من ٥٠٠، ازياد معه الزمن وذلك بسبب أن المتحكمين بحاجة لوقت أكبر لتحديد مسار إرسال الزم نتيجة وجود عدد كبير من المضيفين متصلة بالمبدل نفسه.

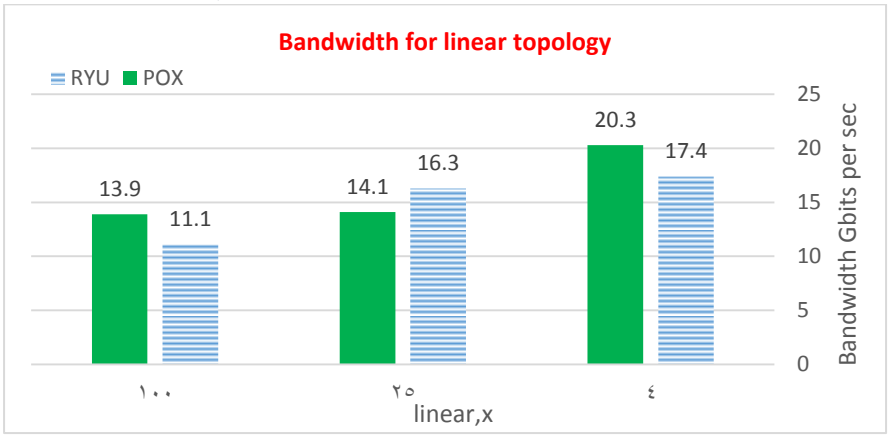
❖ **السيناريو الثالث:** تم إنشاء البنية **Linear** للشبكات المعرفة بالبرمجيات باستخدام RYU وPOX، حيث تتألف البنية من x مبدل وx مضيف، كل مبدل متصل مع مضيف واحد فقط. تمت المقارنة من أجل Linear,4 و linear,25 و linear,100، يمكن بناء البنية عن طريق كتابة السطر:

Sudo mn --controller=remote,ip=127.0.0.1—mac --switch=ovsk,protocols =OpenFlow13 --topo=linear,4

يوضح الشكل (١٢) بنية linear,4 وبالنسبة للبنى ٢٥ و١٠٠ بنفس الطريقة مع ازدياد عدد المبدلات والمضيفين، والشكل (١٣) يمثل عرض الحزمة للبنية الخطية والشكل (١٤) يمثل الإنتاجية للبنية الخطية ، والشكل (١٥) يمثل زمن الذهاب والإياب للبنية باستخدام المتحكمين.

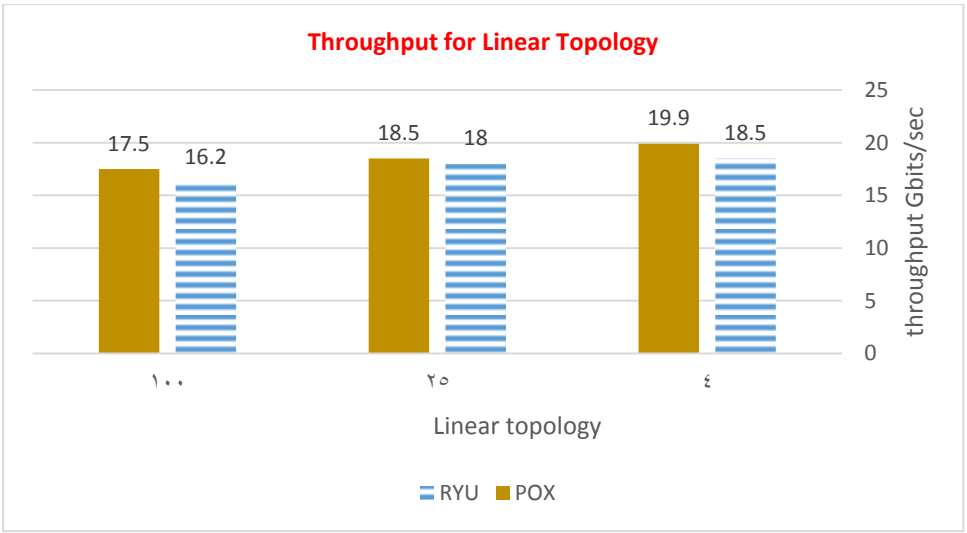


الشكل (١٢) بنية linear,4 في Mininet



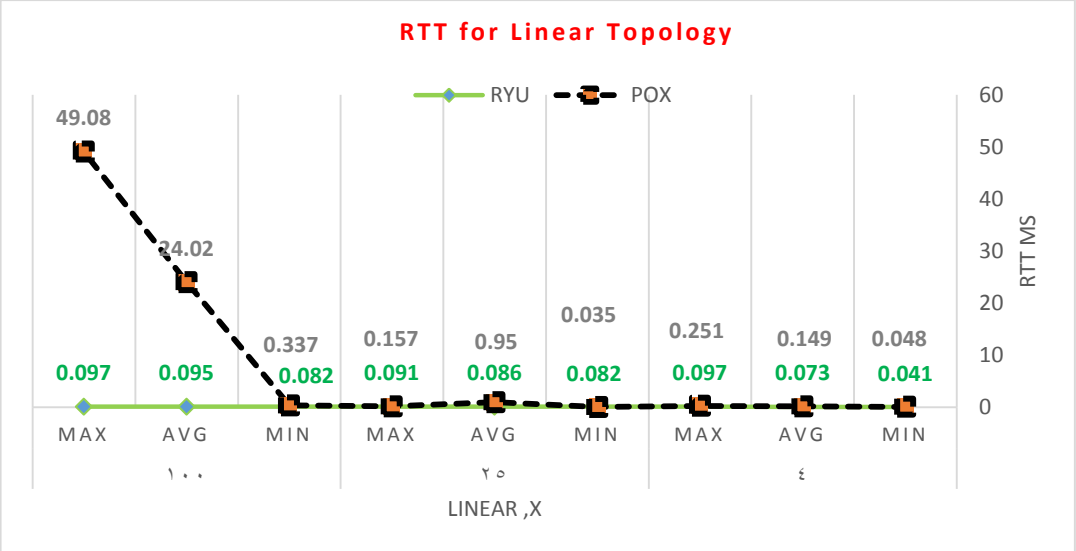
الشكل (١٣): يمثل عرض الحزمة للبنية linear باستخدام المتحكم POX و RYU

يمكن ملاحظة من خلال الشكل (١٣) أن عرض الحزمة عند استخدام المتحكم RYU و POX تناقص مع ازياد عدد المبدلات والمضيفين، حيث في الحالة الأولى لدينا أربع مبدلات وأربع مضيفين فقط وعندما ازداد العدد إلى ٢٥ أدى لتناقص عرض الحزمة ، وتناقص بشكل أكبر عند ازياد عدد المبدلات إلى ١٠٠ مبدل متصلة مع ١٠٠ مضيف والسبب نتيجة الزيادة في عدد المسارات المتشكلة مع كل زيادة في عدد المبدلات وعدم وجود آلية جيدة لعملية توزيع الرزم المرسلة بين المضيفين.



الشكل (١٤): يمثل الإنتاجية للبنية linear باستخدام المتحكم POX و RYU

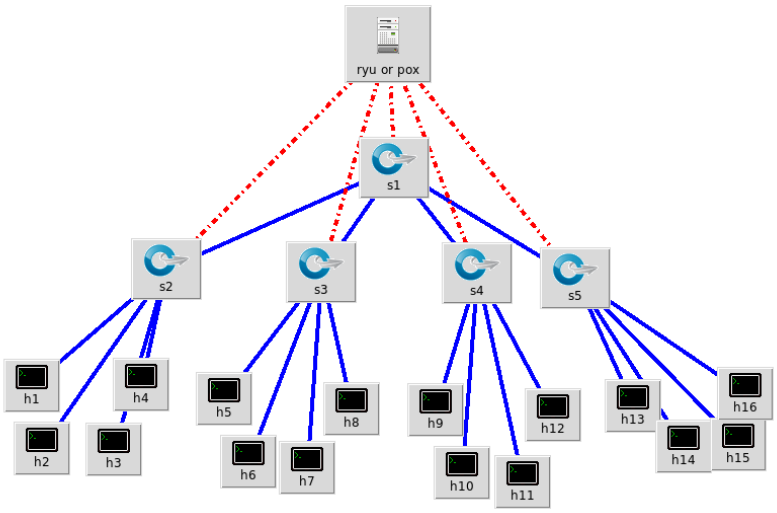
يمكن ملاحظة من الشكل (١٤) الإنتاجية للبنية باستخدام المتحكم RYU بقيت محافظة على قيمها، ويعود السبب أن الإنتاجية لا تتأثر بازدياد عدد المبدلات والمضيفين، وإنتاجية المتحكم POX أفضل بسبب اتباعه آلية أفضل لمعالجة الرزم من المتحكم RYU.



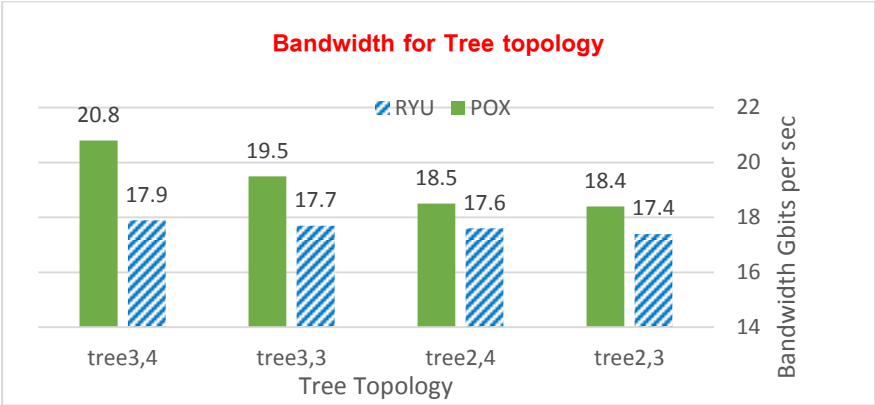
الشكل (١٥): يمثل زمن الذهاب والإياب للبنية linear باستخدام المتحكم RYU وPOX

نلاحظ من الشكل (١٥) أن زمن الذهاب والإياب يحافظ على قيم ثابتة عند استخدام المتحكمين، نلاحظ الزيادة بشكل كبير عند استخدام عدد مبدلات وأجهزة يساوي ١٠٠ ويعود ذلك لأنه يحتاج لوقت أطول لعملية إرسال الرزم واستقبالها.

❖ **السيناريو الرابع:** تم إنشاء البنية الشجرية **tree,x,y** للشبكات المعرفة بالبرمجيات حيث تكون شبكة بعمق x و y أبناء باستخدام المتحكم RYU وPOX. تمت المقارنة من أجل tree2,3 أي switches=4 و host=9، الشبكة الثانية tree2,4 أي switches=٥ و host= 16، الشبكة الثالثة tree3,3 : عدد switch=13 , host=27، الشبكة الرابعة tree3,4: host=64 , switch=21. يوضح الشكل (١٦) البنية الشجرية بعمق ٢ وعدد أبناء ٤، وكانت النتائج كما في الشكل (١٧) و (١٨) و (١٩) لعرض الحزمة والإنتاجية وزمن الذهاب والإياب على التتالي.

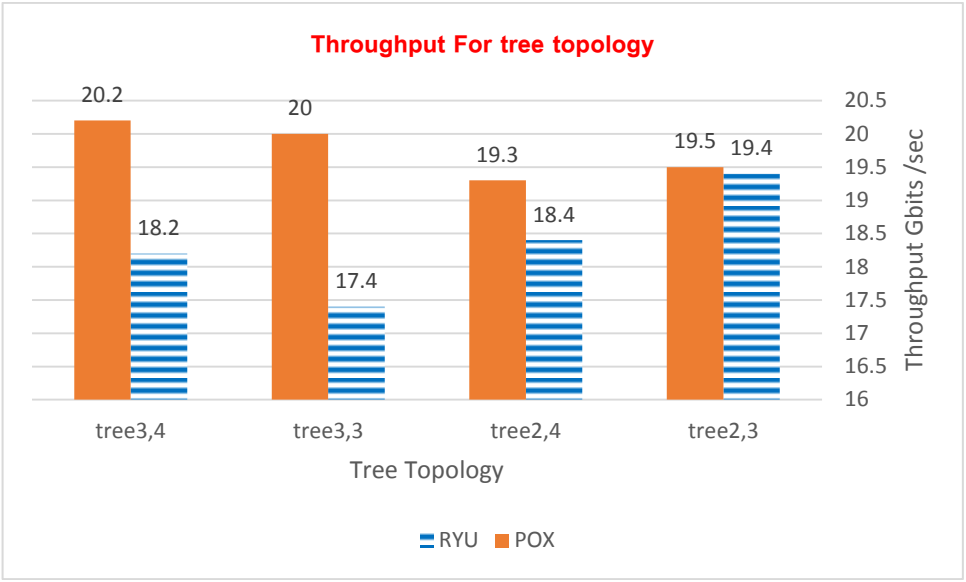


الشكل (١٦) البنية الشجرية بعمق ٢ وعدد أبناء ٤



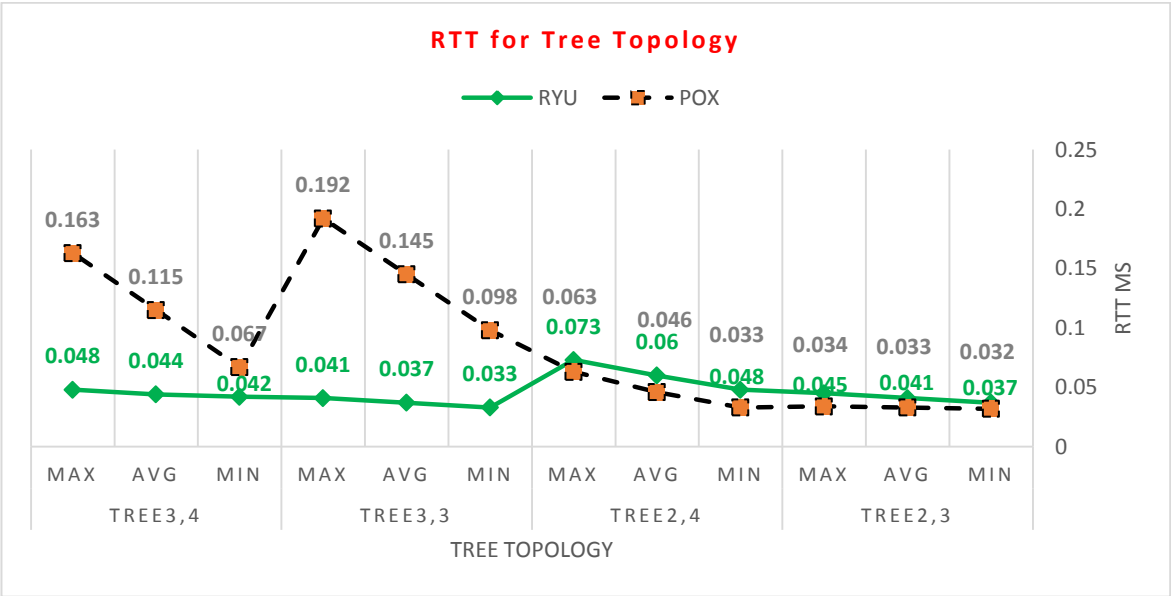
الشكل (١٧): يمثل عرض الحزمة للبنية tree باستخدام المتحكم RYU وPOX

نلاحظ من الشكل (١٧) أن عرض الحزمة يحافظ على قيم متقاربة مع ازياد عدد المبدلات والمضيفين في الشبكة، أيضاً عند استخدام المتحكم RYU يعطي عرض حزمة أقل من قيم عرض الحزمة عند استخدام المتحكم POX، وذلك بسبب امتلاك المتحكم POX آلية أفضل لتوزيع الرزم من المتحكم RYU.



الشكل (١٨): يمثل الإنتاجية للبنية tree باستخدام المتحكم RYU وPOX

نلاحظ من الشكل (١٨) أن قيم الإنتاجية عند استخدام المتحكم POX للبنية Tree تحافظ على قيم متقاربة من بعضها وأعلى من قيم الإنتاجية لنفس البنية عند استخدام المتحكم RYU بسبب قدرته على معالجة الرزم بشكل أفضل من POX. والإنتاجية في البنية Tree3,3 باستخدام RYU منخفضة بسبب القدرة المحدودة للمعالجة وبالتالي احتمالية فقد الرزم أثناء عملية المعالجة.



الشكل (١٩): يمثل زمن الذهاب والإياب للبنية Tree باستخدام المتحكم RYU والمتحكم POX

نلاحظ من الشكل (١٩) أن زمن الذهاب والإياب ثابت في البنى Tree2.4 , Tree2.3 باستخدام المتحكمين لأنهم لا يحتاجوا لوقت طويل لتحديد مسارات إرسال الرزم، أما مع ازياد العمق وعدد الأبناء أدى لازدياد الزمن وذلك بسبب الحاجة لوقت أطول لتحديد مسارات الإرسال للرزم. ويمكن ملاحظة أن المتحكم POX يحتاج زمن أطول من المتحكم RYU بسبب حاجته لوقت أطول لعملية اكتشاف المسارات.

٥-الاستنتاجات والتوصيات:

تم في هذا البحث دراسة أداء الشبكات المعرفة بالبرمجيات باستخدام المتحكم RYU والمتحكم POX من خلال البارامترات (عرض الحزمة والإنتاجية وزمن الذهاب والإياب)، بناء على النتائج التي توصل إليها البحث يمكن استنتاج ما يلي:

١-بالنسبة ل الطوبولوجيا الافتراضية فإن استخدام المتحكم pox أعطى تحسن بالإنتاجية بمقدار ٩.٥% وتحسن بعرض الحزمة بمقدار ١١.٩٩% عن استخدام المتحكم RYU.

٢-عند مقارنة طوبولوجيا Single تبين أن استخدام المتحكم POX أعطى تحسن بالإنتاجية بمقدار ١٨% و١٤.٩% و٢١.٩% و١٨.٨٤% و١٣.٦٩% و١٠.٠١% عن استخدام المتحكم RYU بالإضافة إلى تحسن بعرض الحزمة بمقدار ٥.٦٥% و٦.١٥% و١٥.١٤% و١٢.٨% و٢٠.٦٨% و٤٨.٩٢% للشبكات Single,1000-single,500-single,100 – single,25 – single,8 – single,4 على التتالي.

٣-عند مقارنة الطوبولوجيا Linear أيضا تفوق المتحكم pox بعرض حزمة بنسبة ١٥.٣٨% و١٤.٤٧% و٢٢.٤% وإنتاجية بنسبة ٧.٢٩% و٢.٧٣% و٧.٧١% للشبكات linear,100 – linear,4 – linear,25 على التتالي.

٤-عند مقارنة الطوبولوجيا tree أيضا تفوق المتحكم POX بعرض حزمة بنسبة ٥.٥٨% و٤.٩٨% و٩.٦٧% و١٤.٩٨% وإنتاجية بنسبة ٠.٥١% و٤.٧٧% و١٣.٩% و١٠.٤١% للشبكات tree3,3 – tree3,4 – tree2,3 tree2,4 على التتالي. وبالتالي يمكن القول إن المتحكم POX له تأثير أفضل عند استخدامه في هذه الشبكات على استخدام المتحكم RYU.

٥-بالنسبة لمقارنة أداء الشبكات من خلال زمن الذهاب والإياب يمكن القول: إن الشبكات التي تستخدم المتحكم RYU كان لديها أداء أفضل من الشبكات التي استخدمت المتحكم POX كانت الشبكات الافتراضية لديها أقل زمن وذلك لإن عدد الأجهزة فيها قليل حيث تتألف من مبدل ومضيفين فقط. بالنسبة لبقية الشبكات كانت البنية الشجرية لديها زمن منخفض مقارنة بالشبكات Single و linear وذلك لإن عدد المبدلات والمضيفين متناسبة أكثر من الشبكات الأخرى.

٦-المراجع:

[1] M. CASADO, T. KOPONEN, D. MOON, and S. SHENKER. (2008). *Rethinking Packet Forwarding Hardware.* Seventh ACM Workshop. On Hot Nets-VII.

[٢] ONF, *–SDN Architecture Overview*. (2013) Version 1.0, URL: https://www.opennetwork ing.org/ images/stories/downloads/sun-resources/technical re-ports/ SDN-architecture-overview 1.0. pdf.

[3] Freris, N. M. (2019). A software-defined architecture for control of IoT cyber-physical systems. *Cluster Computing, 22*(4), 1107–1122.

[4] Queiroz, W., Capretz, M. A., & Dantas, M. (2019). An approach for SDN traffic monitoring based on big data techniques. *Journal of Network and Computer Applications, 131*(28–39), 16.

[5] Badotra, S., & Panda, S. N. (2019). Evaluation and comparison of OpenDayLight and open networking operating system in software-defined networking. *Cluster Computing, 1–11*, 17.

[6] Priya, A. V., & Radhika, N. (2019). Performance comparison of SDN OpenFlow controllers. *International Journal of Computer Aided Engineering and*

[7] Bholebawa, I. Z., & Dalal, U. D. (2018). Performance analysis of SDN/OpenFlow controllers: POX versus floodlight. *Wireless Personal Communications, 98*(2), 1679–1699.

[8] Asadollahi, S., Goswami, B., & Sameer, M. (2018). Ryu controller's scalability experiment on software defined networks. In 2018 *IEEE International conference on current trends in advanced computing (ICCTAC)* (pp. 1–5). IEEE.

[9] Eftimie, A., & Borcoci, E. (2020). SDN controller implementation using OpenDaylight: experiments. In *2020 13th International Conference on Communications (COMM)* (pp. 477–481). IEEE.

[10] http://mininet.org, last visited at June 15, 2023.