تقييم أداء بعض المتحكمات في الشبكات المعرفة برمجياً بما يخدم التوسعية والبنية الموزعة

د. أحمد صقر أحمد*

حازم ديب**

(تاريخ الإيداع 24 / 5 / 2021. قُبِل للنشر في 4/ 10 / 2021)

□ ملخّص □

تعتبر الشبكات المعرفة برمجياً Software Defined Networks (SDN) من البنى الشبكية الحديثة والتي تتصف بالمرونة والديناميكية وتوفير الموارد وسهولة الإدارة. تعتبر بنية شبكة SDN كتمثيل لشبكة تم فصل قرارات التحكم فيها عن قرارات التوجيه. يركز هذا البحث على دراسة أداء ثلاث متحكمات رئيسية هي متحكم POX المبرمج بلغة Python، ومتحكم MUL المبرمج بلغة C، ومتحكم OpenDayLight المبرمج بلغة JAVA. تمت عمليات دراسة الأداء لهذه المتحكمات في أربع بنى مختلفة من شبكات SDN، وهي البنية الوحيدة والخطية والشجرية وبنية رابعة قمنا باقتراحها، حيث تركزت عمليات مقارنة الأداء من خلال قياس عاملين هما: الإنتاجية Throughput وزمن الرحلة Round Trip Time. تمت عمليات المقارنة باستخدام المحاكي الشبكي Mininet.

بينت النتائج أن متحكم OpenDayLight يعطي أداءاً أفضل من متحكم POX في جميع البنى المدروسة، و أداءاً أفضل من متحكم MUL في ثلاث من البنى الأربع المدروسة وفق العاملين الذين قمنا بدراستهما وفي البيئات المتنوعة التي تمت الدراسة فيها.

الكلمات المفتاحية: الشبكات المعرفة برمجياً، التوسع، المتحكم، الانتاجية، زمن الرحلة.

*أستاذ – قسم النظم والشبكات الحاسوبية – كلية الهندسة المعلوماتية – جامعة تشرين – اللاذقية –سورية.

Email: ahmad.s.ahmad@tishreen.edu.sy

**طالب دكتوراه – قسم النظم والشبكات الحاسوبية – كلية الهندسة المعلوماتية – جامعة تشرين – اللاذقية – سورية.

Email: hazem.deeb@tishreen.edu.sy

Performance Evaluation of some SDN Controllers Towards Scalable and Distributed SDN

Dr. Ahmad Saker Ahmad*
Hazem Deeb**

(Received 24 / 5 / 2021. Accepted 4 / 10 / 2021)

□ ABSTRACT □

Software-Defined Networking (SDN) is an emerging network architecture that is adaptable, dynamic, cost-effective, and manageable. The SDN architecture is a form of network virtualization where the network controlling and forwarding functions are separated. This research mainly focuses on three controllers, POX, a Python based controller, Mul, a C based controller, and OpenDayLight, a Java based controller. A performance comparison of these controllers is effected over four different SDN network topologies which are: single, linear, tree and a fourth proposed topology. The performance evaluation has been processed by measuring two factors which are: Throughput and Round Trip Time. The performance comparison has been done using the network simulator Mininet.

The results show that the OpenDayLight controller gives better performance than both POX controller in all different topologies, and better than MUL in three of the four tested topologies, and that was according to the two factors in the different environments which were studied.

Keywords: SDN, Scalability, Controller, Throughput, Round Trip Time.

* **Professor, Department of Computer Networks & Systems, Faculty of Informatics Engineering Faculty, Tishreen University, Lattakia, Syria. Email: ahmad.s.ahmad@tishreen.edu.sy**

** **PhD Student, Department of Computer Networks & Systems, Faculty of Informatics Engineering Faculty, Tishreen University, Lattakia, Syria. Email: hazem.deeb@tishreen.edu.sy**

مقدمة:

على عكس الشبكات التقليدية، فإن الشبكات المعرفة برمجياً SDN تقوم باستخدام متحكم مركزي لفصل مستوي التحكم عن مستوي البيانات في الأجهزة الشبكية كالموجهات والمبدلات، وبالتالي فإن شبكات SDN سوف تزيد من عمليات الابداع وقابلية التطوير في مجال الشبكات الحاسوبية. رافق هذا التطور وجود ثلاث مفاهيم أساسية حرجة لا يمكن تحقيقها في شبكات SDN ذات المتحكم المركزي الوحيد وهي: أولاً، أن الفعالية لا تكون كافية باستخدام متحكم مركزي وحيد، ثانياً، قابلية التوسع والتي تعتبر واحدة من أهم القضايا التي تدفع مهندسي الشبكات إلى تطوير بنية شبكات SDN واستبدال المتحكم الوحيد بمتحكمات متعددة، وثالثاً، التوافرية والذي يشمل قضيتين هما المضاعفة والأمن، حيث تعتبر المضاعفة من أهم المفاهيم الموجودة في أي نظام لتجنب نقطة الفشل الوحيدة، والمتحكم المركزي الوحيد من الممكن أن يفشل في أي وقت، ولهذا السبب من الممكن أن تبقى الشبكة بدون متحكم. ومن جهة أخرى، يعتبر مفهوم الأمن من المفاهيم الهامة، فإذا سيطر مهاجم على المتحكم المركزي فستتم خسارة السيطرة على الشبكة وإدارتها بالكامل. لذا من الواضح أنه إذا امتلكنا عدة متحكمات في الشبكة فإننا نستطيع التقليل من هذه المخاطر بشكل كبير لأن هذه المتحكمات سوف تتحد مع بعضها وتحدد فيما إذا كان أحد المتحكمات لا يعمل بشكل صحيح وتقوم بفصله عن الشبكة. كل هذه الأسباب دفعت للتفكير باستبدال المتحكم المركزي الوحيد الموجود في شبكات SDN بعدة متحكمات.

أهمية البحث وأهدافه:

مع التطور الكبير في شبكات SDN، لم تعد البنية التقليدية لهذه الشبكات مفيدة وذلك بسبب ظهور عدد من المشاكل وأهمها نقطة الفشل الوحيدة في المتحكم المركزي، حيث دعا إلى استبدال المتحكم المركزي بعدة متحكمات. ولكن استخدام عدة متحكمات يزيد من تعقيد الشبكات المعرفة برمجياً بشكل كبير، حيث ظهرت تحديات جديدة تتعلق بإمكانية الجدولة والإدارة واختيار المتحكم الأفضل في عمليات الإدارة، لذا تمت المقارنة بين أداء ثلاث أنواع من المتحكمات وهي POX وMUL وOpenDayLight في أربعة بنى مختلفة من شبكات SDN، وذلك لاختيار المتحكم الأفضل بينها في هذه البنى بهدف استخدامه في شبكات SDN ذات المتحكمات المتعددة، مع ملاحظة أن المتحكمات الثلاث المذكورة سابقاً قد تم اختيارها بناءً على بحث سابق وقد تم اختيار المتحكم الأفضل في كل لغة برمجة والمقارنة بينها. من الجدير بالذكر أنه لا يوجد دليل قاطع على أن تميز المتحكم في بنية مركزية تعني بالضرورة تميزه في بنية موزعة، ولكن من خلال قراءة نتائج العديد من الدراسات السابقة وجد أن المتحكم الذي يعطي أداءً عالياً في بنية شجرية مركزية يعطي نتائج أفضل من باقي المتحكمات في البنية الموزعة، وبما أنه كان من الصعوبة بمكان دراسة جميع المتحكمات في بنية موزعة بسبب محدودية الموارد، كان لا بد من مقارنة المتحكمات في بنية مركزية واختيار المتحكم الأفضل بينها واستخدامه في بنية موزعة.

طرائق البحث ومواده:

يتطرق هذا البحث إلى التعرف على مفاهيم التوسع في شبكات SDN، والمخاوف الناتجة عن عملية التوسع هذه، حيث يتم التعرف على البنى المختلفة لشبكات SDN ذات المتحكمات المتعددة، والفوائد والتحديات التي تواجه عملية استبدال المتحكم المركزي بعدة متحكمات، كما تم مقارنة أداء ثلاث أنواع من المتحكمات وهي POX وMUL

وOpenDayLight في أربعة بنى مختلفة من شبكات SDN، وهي البنية الوحيدة والخطية والشجرية وبنية رابعة قمنا باقتراحها، وتم التطرق إلى التجارب التي تم إجراؤها والنتائج التي تم الحصول عليها من خلال محاكي Mininet الذي يعمل على نظام تشغيل Linux.

1- التوسعية في شبكات SDN:

قدمت الشبكات المعرفة برمجياً تطوراً كبيراً في معمارية الشبكات بما يتناسب مع متطلبات التطوير المستمر في عالم الشبكات، لذا تمثل SDN طريقة جديدة في إعداد الشبكة وإدارتها وتحديثها بمرونة وكفاءة عالية، مما يشجع الانتقال نحو استخدامها.

نقل عملية اتخاذ القرار من العناصر الشبكية في طبقة البيانات هو القاسم المشترك الذي يعمل عليه مجتمع الباحثين في شبكات SDN، مما يجلب العديد من الفوائد مثل المرونة العالية، قابلية البرمجة، ورؤية الشبكة ككل من وجهة نظر مركزية. ولكن على الرغم من هذه الإيجابيات كان هنالك دائماً العديد من المخاوف بشأن الأداء وقابلية التوسع منذ نشأة شبكات SDN [1].

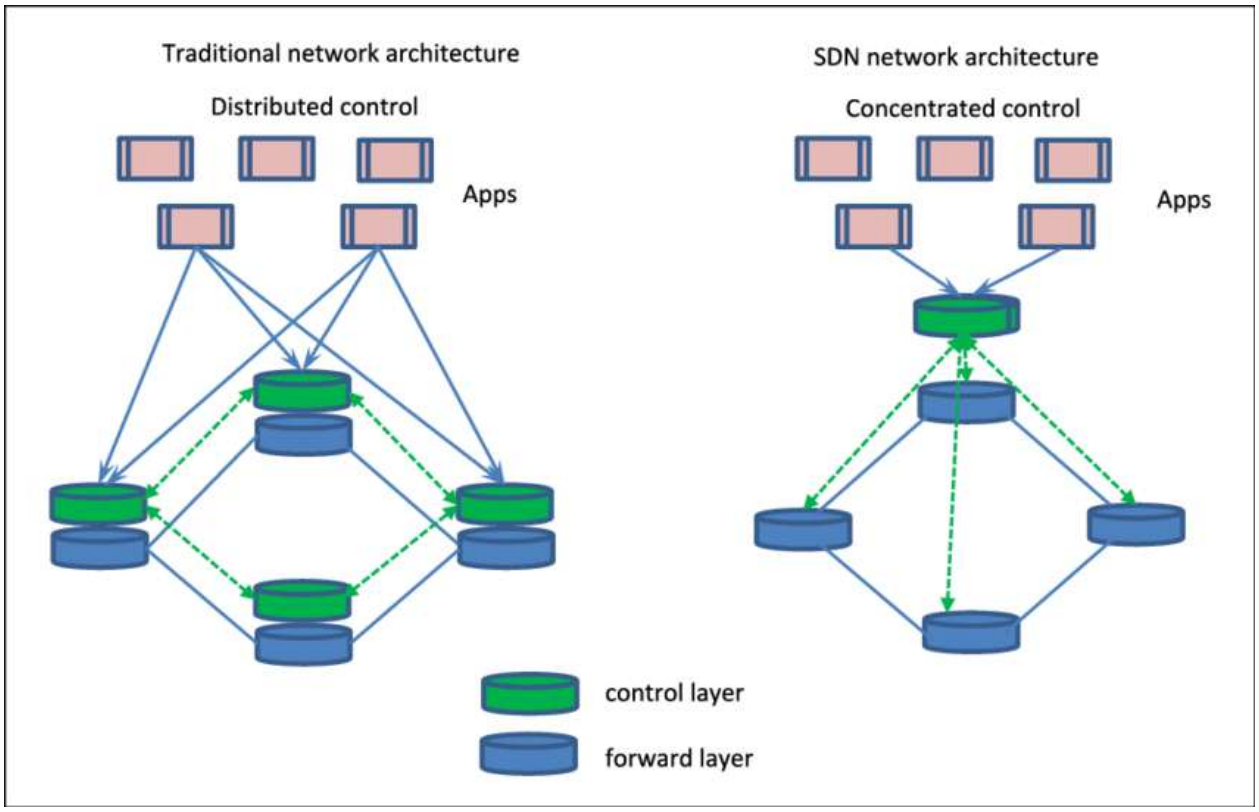
يقود المفهوم المشترك بوجود التحكم بشكل مركزي في شبكات SDN إلى وجود مخاوف عديدة متعلقة بالمرونة وقابلية التوسع، فمهما بلغت إمكانيات المتحكم المركزي، لا يوجد متحكم واحد يستطيع أن يواكب النمو في الشبكة (زيادة عدد المبدلات، التدفقات، عرض الحزمة، وغيرها..)، وسوف يفشل هذا المتحكم في معالجة كل الطلبات القادمة إليه، وفي الاستمرار بتقديم نفس المستوى من جودة الخدمة. أثبتت دراسات سابقة على متحكم NOX، وهو أول متحكم SDN، أنه يستطيع معالجة 30000 تدفق فقط كل ثانية، مما يضطرنا إلى تخفيض معدل التدفق، وذلك تجنباً لفشل المتحكم، ولكن مع تخفيض معدل التدفق تظهر مخاوف جديدة بشأن التأخير وزيادة الوقت اللازم لمعالجة الطلبات. من أجل تجنب هذه المخاوف التي قد تسبب العديد من المشاكل، أهمها مشكلة عنق الزجاجة كان لا بد من التخلي عن المفهوم التقليدي لشبكات SDN بوجود متحكم مركزي والانتقال إلى مفاهيم موزعة بشكل أكبر[2].

2- مفاهيم التوسعية:

فصل مستوي التحكم عن مستوي البيانات هو العامل المشترك الرئيسي في شبكات SDN، وهو ما تختلف به عن الشبكات التقليدية السابقة لها، يقود هذا إلى خصائص مهمة، لعلَّ أهمها هو القدرة على تطوير كل من طبقة التحكم وطبقة البيانات بشكل مستقل عن الآخر، بشرط وجود واجهة برمجة تطبيقات API، حيث يقوم كل من مستوي التحكم ومستوي البيانات بتبادل المعلومات بينهما عن طريق هذه الواجهة كما هو موضح في الشكل(1). تتكون العناصر الشبكية الموجودة في طبقة البيانات والمسؤولة عن عملية تمرير البيانات من مجموعة من عناصر الدارات المتكاملة الخاصة بالتطبيقات، والتي تتطور بشكل أبطأ من العناصر المشكلة لطبقة التحكم والتي هي عبارة عن برمجيات. بالإضافة إلى أن وجود رؤية مركزية للشبكة ككل يخلق فرصاً هائلة لتبسيط تطبيقات التحكم لتسهيل عمليات التغيير والإبداع في مستوي التحكم[3].

على الرغم من كل الإيجابيات السابقة فإن الفصل له عيوبه أيضاً، فتعريف واجهة برمجة تطبيقات موحدة للتخاطب بين كل من طبقتي التحكم والبيانات ليس بالأمر البديهي، وهو ما يستدعي أن تكون هذه الواجهة قابلة للعمل مع اختلاف البنى في شبكات SDN، ويجب أن تكون أيضاً قادرة على تسهيل التطوير لكل من طبقتي التحكم والبيانات بشكل مستقل، كما يتوجب على جميع مصنعي المبدلات والأجهزة الشبكية استخدام نفس هذه الواجهة الموحدة، وإلا فقد يؤدي ذلك إلى تقييد الشبكات بحيث لا تعمل إلا الشبكات التي تعتمد على مصدر واحد لأجهزتها الشبكية، ما سيبطئ من

الكثير من عمليات المرونة والتوسع والإبداع، كما أن هذا الفصل قد يقودنا إلى مشاكل متعلقة بإمكانية التوسع، فنقل عملية اتخاذ القرار إلى متحكم بعيد قد ينتج أماكن وأنواع جديدة من الاختناقات في الشبكة تتعلق بطبيعة وبنية الشبكة والتطبيقات الخاصة بها. ويوضح الشكل (1) الفرق بين الشبكات التقليدية والشبكات القابلة للبرمجة[4].



الشكل (1) توزع شبكات SDN بالمقارنة مع الشبكات التقليدية

3- توسع المتحكمات:

يعتبر توسع المتحكمات هو النقطة الرئيسية في إمكانية توسع شبكات SDN، حيث أن أهم نموذج في هذه الشبكات هو تصميم شبكة بمتحكم مركزي وحيد، ومع ازدياد أعداد المستخدمين فإن عدداً أكبر من الطلبات ترد إلى المتحكم المركزي، وفي لحظة معينة يصبح هذا المتحكم غير قادر على معالجة هذه الطلبات مما يعرضه للفشل، وكما ذكرنا سابقاً فإن متحكم NOX يمكنه معالجة 30000 طلب كل ثانية، وهذا قد يكون مناسباً لشبكة محدودة ولكنه غير فعال في مراكز البيانات الكبيرة التي تتطلب تدفقاً عالياً من البيانات. أحد الحلول لهذه المشكلة هو إجراء بعض التعديلات على المتحكم مما قد يساعد بتسريع أدائه بالشكل الذي يسمح له بدعم شبكات أكبر مع عرض حزمة وتأخير مناسبين، أما الحل الآخر فيكون من خلال وضع مجموعة من السياسات لتقليل الطلبات الواردة إلى المتحكم، أو باستخدام متحكم وحيد مع وجود عدة نوى وعدة واجهات لبرمجة التطبيقات، حيث تقوم على توزيع الطلبات على نوى المتحكم المركزي. لكن يبقى الحل الأفضل من خلال وجود متحكمات متعددة في الشبكة، وهنا تبرز مشكلة المزامنة بين هذه المتحكمات وتوزيعها في الشبكة، حيث من الممكن تقسيمها إلى عدة مستويات مثل المتحكمات المحلية، والتي تكون قريبة من طبقة البيانات وتكون مسؤولة عن معالجة الطلبات المحلية المتكررة (الطلبات التي لا تحتاج إلى رؤية عامة للشبكة)، وتحمي

المستويات الأعلى من طبقة التحكم من زيادة الحمل، بينما يقوم المتحكم في الطبقة الأعلى بالتعامل مع الطلبات التي تحتاج إلى رؤية عامة للشبكة كما يقوم بدور التنسيق بين المتحكمات المحلية[5].

4- المخاوف الناتجة عن التوسع:

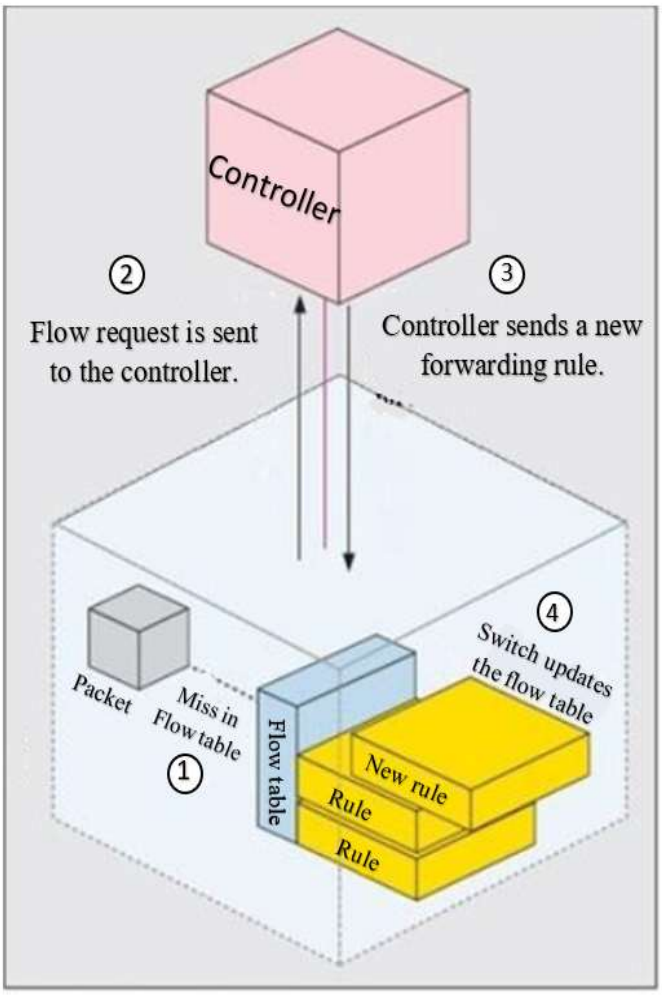
يعتبر زيادة الحمل على المتحكم أحد المخاوف التي تجعلنا بحاجة إلى التوسع ولكن يوجد مخاوف أخرى مثل:

4-1- المخاوف المتعلقة بتدفق البيانات:

للتعرف على هذه المخاوف يجب أن نعرف آلية العمل أثناء تدفق البيانات والموضحة في الشكل (2) :

1) تصل حزمة جديدة ذات مدخل غير معرف في جدول التدفق.
2) يقوم المبدل بإنشاء طلب جديد إلى المتحكم .
3) يقوم المتحكم باتخاذ القرار وإنشاء قاعدة جديدة لهذا الطلب ويقوم بإرسالها إلى المبدل.
4) يقوم المبدل بالتنفيذ وتحديث جدول التدفق الموجود لديه.

يعتمد الأداء في الخطوات 4،2،1 على إمكانيات المبدل (وحدة المعالجة، الذواكر)، بينما يعتمد في الخطوة 3 قبل الأخيرة على إمكانيات المتحكم، وبما أن المتحكمات موجودة جغرافياً في أماكن قريبة من المبدلات فإن زمن التأخير الناتج عن عملية تبادل المعلومات بين المبدلات والمتحكمات يمكن إهماله، وبالتالي تصبح المخاوف بشأن الإمكانيات التي يجب أن تكون موجودة في كل من المتحكمات والمبدلات في الشبكة. يوضح الشكل (2) آلية تدفق البيانات في شبكات SDN.



الشكل (2) خطوات تدفق البيانات في شبكات SDN

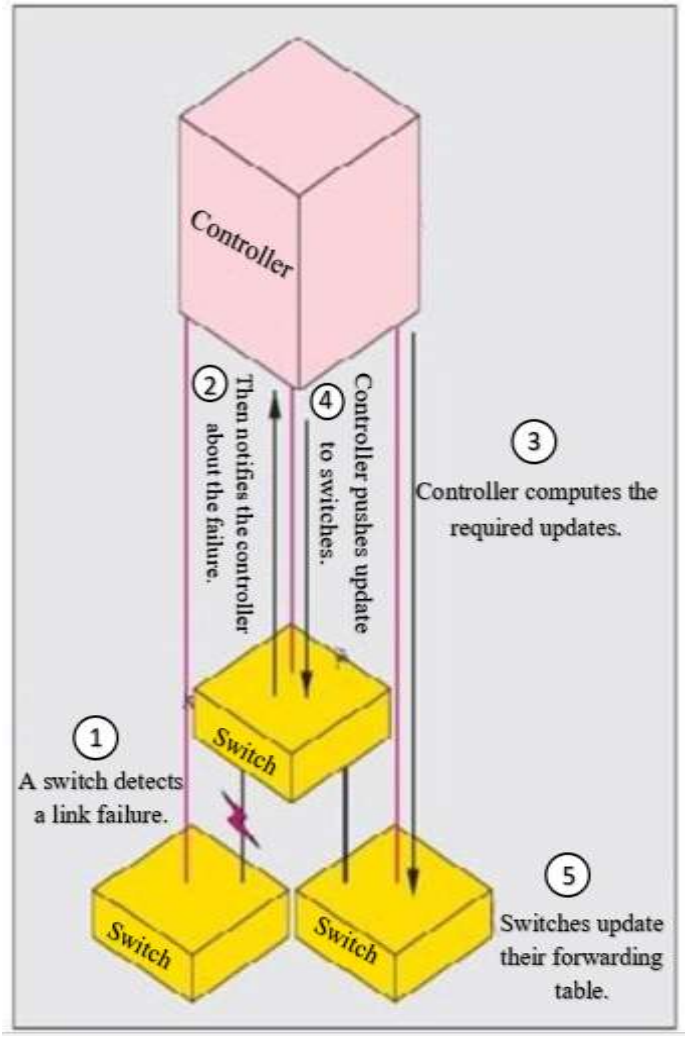
على الرغم من أن المتحكمات تستطيع أن تعالج الآلاف من التدفقات في الثانية مع تأخير زمني صغير نسبياً، ماتزال المبدلات الشبكية محدودة الأداء، حيث تستطيع معالجة مئات التدفقات في الثانية مع تأخير زمني مقدر ب 10 msec في أفضل حالاتها، وهذا الأداء لا يزال يعتبر ضعيفاً نسبياً بسبب ضعف موارد المبدلات مقارنة مع نمو الشبكات والبرمجيات الخاصة بالمتحكمات، لذا كان لا بد من العمل على تطوير عتاديات المبدلات.

4-2- المخاوف من الفشل في أحد الوصلات:

تعتبر المرونة في معالجة الفشل في أحد الوصلات عاملاً مهماً في الشبكات التقليدية، وشبكات SDN ليست استثناءاً، فعند حدوث فشل في أحد وصلات شبكة SDN، قد تفقد أكثر من نصف الشبكة اتصالها بالمتحكم المركزي، ولكن مع وجود أكثر من متحكم في الشبكة، ستقوم المبدلات بمحاولة اكتشاف متحكم آخر موجود في قسمها لتتصل به وتستمر عملية معالجة التدفقات. وستقوم بالتعرف على آلية معالجة الفشل في أحد الوصلات أو المبدلات في شبكات SDN في الشكل (3)، وتتم هذه العملية بخمس خطوات:

1) يكتشف المبدل حدوث فشل في إحدى الوصلات.
2) يقوم هذا المبدل بإخطار المتحكم بهذا الفشل.
3) يقوم المتحكم بحساب عمليات الإصلاح التي يمكن تنفيذها.
4) يقوم بإرسال التحديثات التي تمت إلى المبدل.
5) يقوم المبدل بدوره بتحديث الجداول الموجودة لديه.

تختلف هذه العملية عن عملية المعالجة في الشبكات التقليدية، فعند اكتشاف فشل في وصلة ما في الشبكات التقليدية يتم تعميم هذا الفشل على كافة عقد الشبكة، بينما في شبكات SDN يتم إخطار المتحكم فقط والذي يقوم بمعالجة هذا الخطأ وإرسال التحديثات اللازمة، بالتالي فإن التأخير الناتج عن تبادل المعلومات في شبكات SDN هو أفضل حتماً من الشبكات التقليدية. يوضح الشكل (3) خطوات معالجة الفشل في أحد الوصلات في شبكات SDN.



الشكل (3) خطوات معالجة الفشل في أحد الوصلات في شبكات SDN.

وعلى الرغم من أن مفهوم معالجة فشل وصلة ما في شبكات SDN متطور أكثر من الشبكات التقليدية، ولكن لا تزال المخاوف موجودة حيث يجب أن توفر هذه المتحكمات طرقاً بديلة في كل من المبدلات، وذلك كي يقوم المبدل باستخدام هذه الطرق في حال اكتشاف خطأ ما بدلاً من أن يقوم بإرسال رسالة خطأ إلى المتحكم في كل مرة يتم فيها اكتشاف فشل في وصلة ما[6].

5- التوسع من مفاهيم أخرى في مختلف مجالات الشبكة:

المتحكم الوحيد غير قادر على معالجة تدفقات البيانات الضخمة في شبكات SDN الكبيرة نسبياً، مما يسبب مشاكل عديدة خاصةً في حال الحاجة إلى التوسع في بنية الشبكة، لذا كان لا بد من اللجوء إلى البنية الموزعة في شبكات SDN. وقمنا لحد الآن بالتعريف بأهم المخاوف المتعلقة بالتوسع في شبكات SDN من خلال التعرف على العديد من المعايير التي تؤثر على نمو الشبكات، ولكن سوف نتطرق في هذا القسم إلى مفهوم جديد من خلال شرح مفاهيم التوسع في الأنواع المختلفة من الشبكات، حيث اخترنا شبكات مراكز البيانات وشبكات مزودي الخدمة كأمثلة، وذلك لأن كل منها يتطلب إعدادات مختلفة، وليس الهدف هنا هو تعداد كافة أنواع الشبكات، ولكن الهدف هو التعرف على بعض النماذج التي يمكن تعميمها واستخدامها كمراجع في قضايا التوسع للشبكات الأخرى غير النوعين المذكورين[7].

5-1- شبكات مراكز البيانات:

إن شبكة مراكز البيانات التقليدية تملك عادة عشرات الآلاف من العقد الشبكية والتي يمكن أن تنمو بشكل كبير في وقت قصير نسبياً، مما يعني زيادة عدد الطلبات الواردة إلى المتحكم بشكل كبير يؤدي إلى إغراقه وإيقاف عمله. لذا كان لابد من اتخاذ مجموعة من الخطوات وأهمها وضع مجموعة سياسات تقلل من الطلبات الواردة إلى المتحكم قدر

الإمكان، وأن يتم معالجة الطلبات قبل وصولها إلى المتحكم. عندما يريد أحد التطبيقات عملية معالجة معينة على المستوى المحلي يمكن للمتحكم تفويض المبدلات القريبة من هذا الطلب بمعالجته دون الرجوع إلى المتحكم المركزي، وتكرار هذه العملية مع الطلبات المتكررة على المستوى المحلي، وبذلك يتم تقليل الحمل عن المتحكم المركزي[8].

5-2- شبكات مزودي الخدمة:

تكون شبكات مزودي الخدمة عادةً أصغر من شبكات مراكز البيانات وتملك عدداً أقل من العقد الشبكية، ولكن المشكلة الرئيسية في هذا النوع من الشبكات أنها تكون موزعة جغرافياً بشكل كبير، يثير هذا الامتداد الجغرافي الكبير مخاوف تتعلق بالثبات وضبط التدفقات والتأخير الزمني الناتج عن عملية تبادل البيانات، ولكن يمكن الاستفادة من هذا الامتداد في تقسيم الشبكة بشكل فيزيائي إلى مناطق جغرافية منفصلة بحيث يمكن التحكم بكل قسم من خلال متحكم مركزي منفصل، وتقوم هذه المتحكمات المركزية بتبادل المعلومات الضرورية فقط فيما بينها، ويتم إخفاء معظم الطلبات عن المتحكمات الخارجية. بالإضافة إلى زمن الاستجابة العالي فإن تدفق البيانات في شبكات مزودي الخدمة يكون بمعدل أكبر من الأنواع الأخرى من الشبكات، وبالتالي فإن إمكانيات وحجم قاعدة البيانات الخاصة بكل متحكم أو مبدل تعتبر أمراً يجب أخذه بعين الاعتبار[9].

6- الإمكانيات والتحديات:

تم بشكل عام دراسة إمكانية التوسع من قبل عدد من المفاهيم التي تعتمد على الأداء ولكن ثمة مفاهيم أخرى تؤثر في عمليات التوسع في الشبكة أهمها:

- **الإدارة:** يجب أن تكون إمكانية الإدارة سهلة وقابلة للتطبيق عند وجود أي تعديل في الشبكة كإضافة أو حذف عناصر أو تعديلها، وإن السهولة في عملية التعرف على التعديل الذي طرأ، وسهولة تطبيقه ووضعه في الخدمة تعتبر مفهوماً أساسياً في عملية التوسع .
- **التحقيقات البرمجية:** تملك جميع متحكمات SDN واجهات لتحقيق الوظائف في واجهة برمجة التطبيقات المستخدمة للتخاطب بين طبقة التحكم وطبقة البيانات، هذه الواجهات مطلوبة لكتابة وتحقيق المتحكم ولكن ليس من الضروري أن تستخدم في عمليات إدارة الشبكة ككل، وإن معظم التفاصيل والتعقيد يجب أن تتم في المتحكم بعيداً عن واجهات الإدارة التي تكون مسؤولة عن إدارة عمليات النمو في الشبكة.
- **التجربة والتحقق:** بدون وجود الأدوات اللازمة فإن حل أي خطأ يصبح مشكلة كبيرة في أي نوع من أنواع الشبكات وخاصة مع الشبكات التي تنمو بسرعة، وهذا ينطبق تماماً على شبكات SDN. وبالرغم من أن الأدوات الموجودة لحل المشاكل في شبكات SDN هي أكثر تطوراً من أي نوع من الشبكات الأخرى ولكن تقليص الفجوة بين هذه الأدوات والعمل المطلوب منها يعد تحدياً كبيراً في الوقت الراهن.
- **التمدد:** شبكات SDN في عصرنا الحالي تدعم فقط عدداً معيناً من البروتوكولات المعروفة والشائعة ولكن مع التطور الكبير الذي نشهده فإن نسخاً جديدة من البروتوكولات المعروفة وبروتوكولات جديدة هي قيد الظهور وبالتالي يجب تطوير شبكات SDN بالشكل الذي يجعلها تدعم هذه البروتوكولات قيد الظهور[10].

7- بنية شبكات SDN بوجود عدة متحكمات:

إن الهدف من وجود عدة متحكمات العمل جنباً إلى جنب بهدف تحسين أداء شبكات SDN، بالإضافة إلى جعل إمكانية التوسع أسهل، ولذلك سوف نتعرف فيما يلي على الفرق بين الشبكات المركزية والموزعة، المنطقية والفزيائية، المسطحة والهرمية[11] .

7-1- البنية الفيزيائية المركزية والبنية الفيزيائية الموزعة Physically centralized vs Physically Distributed:

في شبكات SDN، هناك نوعان رئيسيان من الممكن أن تكون عليها بنية الشبكة، إما موجودة فيزيائياً بشكل مركزي أو موزعة فيزيائياً. كان التوجه في بدايات شبكات SDN إلى البنية المركزية ولكن بالنظر إلى عدد من المشاكل مثل نقطة الفشل الوحيدة وإمكانية التوسع، اقترح العلماء الانتقال إلى البنية الموزعة، ومن المهم أن نعرف أننا نتحدث عن شبكات بعدة متحكمات فقط في البنية الفيزيائية الموزعة. استخدمت متحكمات مثل NOX و Beacon تقنيات تعدد النياسب من أجل تحسين أداء المتحكم المركزي الوحيد ولكن بقينا نتعامل مع متحكم وحيد مركزي ولم نستطع أن نحل عدداً من المشاكل الموجودة مثل نقطة الفشل الوحيدة. تختلف البنية الموزعة عن المتحكم المركزي بعدة نياسب بشكل كبير وذلك في مجالات مختلفة مثل إمكانية توزيعها على مناطق أوسع والاتصالات التي تتم بين هذه المتحكمات وإمكانية تجنبها للمشاكل الموجودة في المتحكم المركزي [12].

7-2- البنية المنطقية المركزية و البنية المنطقية الموزعة Logically Centralized vs Logically Distributed:

يمكن أن تكون البنية الموزعة الفيزيائية مركزية منطقياً أو أن تكون موزعة منطقياً. يمكن أن تستغل البنية المركزية منطقياً الميزات الموجودة في البنى ذات المتحكمات المتعددة ولكنها في نفس الوقت تعتبر أن هناك متحكماً مركزياً واحداً، أي تقوم بوضع عدة متحكمات ولكن جميع هذه المتحكمات تملك نفس المعلومات ونفس المسؤوليات وذلك عن طريق عمليات تبادل الرسائل والمزامنة التي تقوم بها الشبكة، ويكون كل متحكم هو متحكم مركزي بالنسبة للمبدلات التي تتصل به، وتعاني هذه التقنية من عدد من المشاكل أهمها الحمل الزائد على الشبكة، بالإضافة إلى مشكلة اختيار البديل في حال فشل أحد المتحكمات. أما في البنية الموزعة منطقياً فإن المتحكمات تكون موزعة بشكل فيزيائي وبشكل منطقي، وكل متحكم يكون مسؤول عن جزء من الشبكة فقط، ويمكنه من اتخاذ القرارات المتعلقة بهذا الجزء فقط من منظوره إلى هذا الجزء بينما في البنية المركزية منطقياً يكون اتخاذ القرارات اعتماداً على منظور المتحكم للشبكة ككل.

باختصار فإن البنية المنطقية المركزية قد حافظت على المفهوم الرئيسي لشبكات SDN بوجود متحكم مركزي يقوم باتخاذ القرارات بناءً على رؤيته للشبكة ككل، بينما البنية المنطقية الموزعة خرجت عن هذا المفهوم بوجود عدة متحكمات كل منها بمسؤوليات مختلفة ومسؤولة عن اتخاذ القرارات فقط في الأجزاء المسؤولة عنها[13] .

7-3- البنية المسطحة و البنية الهرمية Flat vs Hierarchical :

في البنية المسطحة أو الأفقية تكون المتحكمات موجودة على مستوي واحد أو بكلمات أخرى فإن طبقة التحكم تكون مؤلفة من طبقة واحدة ويكون كل متحكم يملك نفس المسؤوليات في نفس الوقت مع باقي المتحكمات ويملك رؤيته الخاصة بالشبكة ككل. أما في البنية الهرمية تكون المتحكمات موزعة بشكل عمودي أي أن طبقة التحكم تتألف من عدة طبقات عادة 2 أو 3 طبقات، وهنا يوجد لكل متحكم مسؤوليات مختلفة ويمكنه من اتخاذ القرارات بناءً على وجهة نظره للجزء المرتبط به من الشبكة .

تملك هاتين الطريقتين عدداً من الإيجابيات والسلبيات فكلاهما يحسنان من سرعة الاستجابة ويعطيان أداءً أفضل من الشبكات ذات المتحكم المركزي الوحيد، في البنية المسطحة يمكن معالجة الأخطاء بشكل أسهل ولكن في نفس الوقت

فإنها صعبة الإدارة. بينما في البنية الهرمية إمكانية الإدارة أسهل بالمقارنة مع البنية المسطحة ولكن ما تزال تعاني من مشكلة نقطة الفشل الوحيدة وذلك في حال حدوثها في الطبقة العليا من طبقة التحكم[14].

8- فوائد استخدام متحكمات متعددة في شبكات SDN :

يجب أن يتم تقسيم شبكات SDN الكبيرة إلى عدة مجالات، وكل مجال يحتاج إلى متحكم واحد على الأقل، وفي مثل هذا السيناريو فإن وجود متحكمات متعددة ضروري لإدارة الشبكة ككل. وجود عدة متحكمات يمكن أن يزيد الاستقرار في الشبكة لأنه يضمن أن تتابع المبدلات عملها حتى في حال فشل أحد المتحكمات أو أحد الوصلات. يمكن تحقيق وجود عدة متحكمات بطريقة موزعة فيزيائياً لكن مركزية منطقياً بالشكل الذي يحقق أهم الفوائد من استخدام متحكمات متعددة وهي:

- **سهولة الإدارة:** تعتبر سهولة الإدارة من أهم الميزات التي توفرها شبكات SDN بعدة متحكمات، حيث أن قدرات المتحكم المركزي ضئيلة بالمقارنة مع وجود متحكمات متعددة، ومن الصعب على متحكم مركزي وحيد أن يقوم بإدارة شبكة كبيرة تتألف من قطاعات مختلفة، وذلك بسبب الخصائص المختلفة لكل من المبدلات الموجودة في القطاعات المختلفة بينما يكون هذا الأمر سهلاً في حال وجود متحكمات متعددة.
- **إمكانية التوسع:** إن التطور الحاصل في بروتوكول OpenFlow سيؤدي إلى زيادة الحمل على المتحكم المركزي الوحيد الموجود في شبكات SDN مما يقلل من إمكانية التوسع فيها، ولذلك كان الحل هو بوجود عدة متحكمات بحيث يمكن إضافة أو إزالة متحكم بسهولة وبشكل ديناميكي مما يجعل قابلية التوسع أسهل بكثير، كما يمكن استخدام تقنيات توزيع الحمل للطلبات الواردة على المتحكمات المتعددة مما يجنبنا حدوث مشكلة عنق الزجاجة على أحد المتحكمات.
- **تقليل زمن الاستجابة:** هناك العديد من التطبيقات الحساسة للوقت والتي تتعامل مع المتحكمات فإذا كان كل طلب يتم الرد عليه من قبل متحكم مركزي بشكل متتالي فإن هذا المتحكم المركزي سوف يصبح ذو حمل زائد وقد يتعرض للفشل أو تأخير كبير في زمن الاستجابة، بينما في حالة المتحكمات المتعددة يمكننا استخدام تقنيات توزيع الحمل في معالجة الطلبات الواردة وارسال الطلب إلى المتحكم المناسب له أي الأقل حملاً مما يقلل من الحمل وزمن الاستجابة بشكل كبير .
- **المتانة:** بنية شبكة SDN بمتحكمات متعددة تملك عدداً من نقاط الاستعادة وذلك لتجنب مشكلة نقطة الفشل الوحيدة مما يزيد من متانة الشبكة، ففي حال فشل أحد المتحكمات تستطيع باقي المتحكمات، بتعديل بسيط أو بدونه من توزيع حمل المتحكم الذي تعرض للفشل بينها وتحافظ بذلك على استقرار الشبكة وأدائها، كما أن وجود عدة متحكمات يقلل بنسبة كبيرة من حدوث أخطاء في الشبكة وخاصة تلك المتعلقة بالمتحكمات[15].

9- التحديات التي تواجه استخدام متحكمات متعددة في شبكات SDN:

بالرغم من أن بنية شبكات SDN ذات المتحكمات المتعددة جاءت كحل لمشكلة الشبكات ذات المتحكم المركزي إلا أنها قد واجهت بعض المشاكل والتحديات التي يجب حلها، ولعل أهمها:

- **الثبات:** عند وجود عدة متحكمات فإن مشكلة مزامنة حالة الشبكة ومعلوماتها بين هذه المتحكمات تعتبر مشكلة معقدة وذلك لأنه يجب أن تبقى هذه المتحكمات متزامنة فيما بينها دون زيادة الحمل على الشبكة رغم زيادة

الرسائل المتبادلة بين هذه المتحكمات، وهذا ما يجعل مشكلة الثبات من أهم التحديات التي تعترض شبكات SDN ذات المتحكمات المتعددة.

- **التوضع:** عند تطوير شبكات SDN باستخدام متحكمات متعددة يجب الإجابة على سؤالين رئيسيين هما: كم من المتحكمات تحتاج هذه الشبكة؟ وما هو الموقع المناسب لكل متحكم في الشبكة؟

كلا السؤالين من تعقيد NP-Hard أي أنه لا يمكن ضمان إيجاد حل أمثلي في زمن خطي، ولكن هذين السؤالين يلعبان دوراً أساسياً في الأداء الذي تقدمه شبكات SDN ذات المتحكمات المتعددة.

- **الجدولة:** إن أداء شبكات SDN سوف يتطور بشكل كبير عند استخدام متحكمات متعددة ولكن تبقى مجموعة من القضايا العالقة مثل: كيف سنقوم بعمليات الجدولة وذلك لمنع أحد المتحكمات من أن يتعرض لحمل زائد؟ وإذا تعرض أحد هذه المتحكمات لحمل زائد ما هي سياسة الجدولة المتبعة لإعادته إلى الوضع الطبيعي بأسرع وقت ممكن ؟ لذا يجب أن نقوم بتعديل على طبقة التحكم لتعالج هذه المجموعة من القضايا، وهذا ما يعتبر تحدياً آخر في هذا النوع من الشبكات [16].

النتائج والمناقشة:

1- السيناريو المقترح:

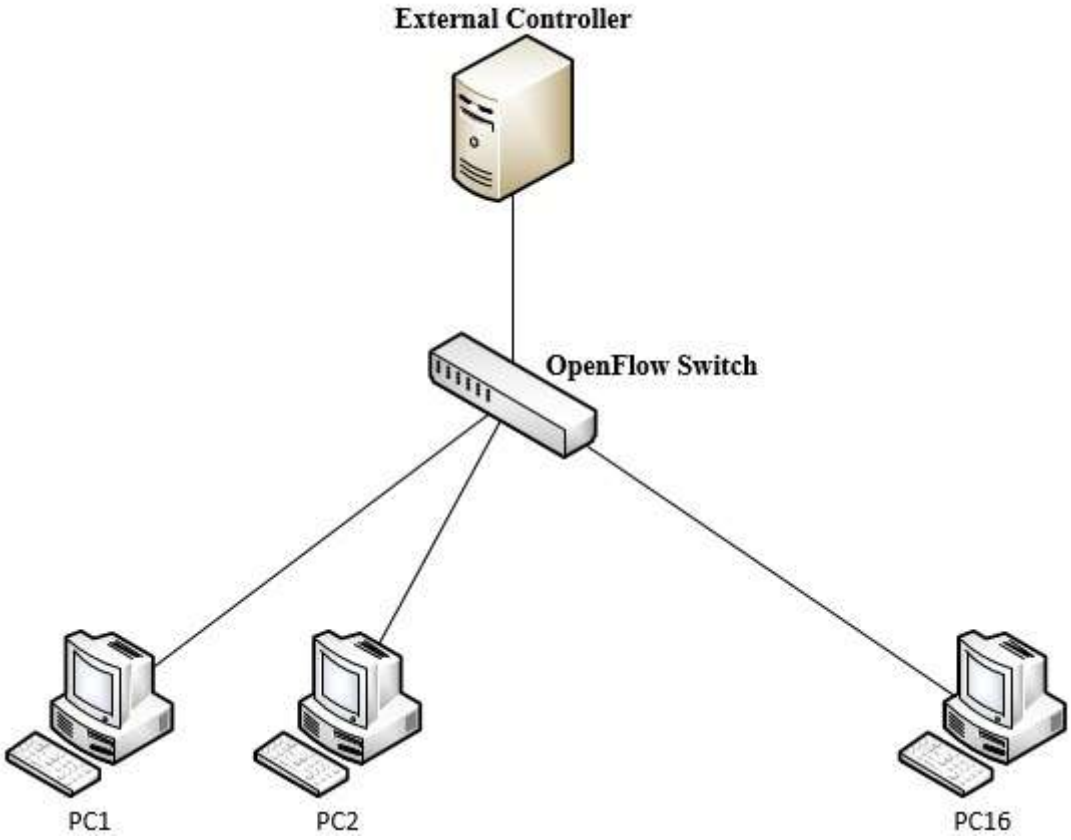
تم مقارنة أداء ثلاث أنواع من المتحكمات وهي POX و MUL و OpenDayLight في أربعة بنى مختلفة من شبكات SDN وهي البنية الوحيدة والخطية والشجرية وبنية رابعة قمنا باقتراحها، تمت عمليات مقارنة الأداء من خلال قياس عاملين هما الإنتاجية Throughput وزمن الرحلة Round Trip Time. حيث أن الإنتاجية تعرف بأنها كمية البيانات التي تم تسليمها خلال وحدة معينة من الزمن، وزمن الرحلة يتم حسابه عن طريق قياس الزمن بين الرسالة Echo Request والرسالة Echo Reply الخاصة بها. تتميز البنية التي قمنا باقتراحها بالديناميكية وإمكانية التوسع، حيث تسمح بإضافة عدد كبير من المتحكمات وأجهزة المضيفين، مما يسهل عملية محاكاة شبكات SDN كبيرة وتجربة سيناريوهات مختلفة، وخاصةً تلك التي تتعلق بشبكات SDN الموزعة.

الأداة التي قمنا باختبار التجارب عليها تمتلك معالج Intel® Core™ i5-2430M CPU @ 2.40 GHZ ، وذواكر بسعة 8 GB DDR3 بنظام تشغيل Windows 10 مع أداة افتراضية على برنامج Virtual Box تم عليها تنصيب نظام تشغيل Linux 18.01 – 64 bit .

2- تحليل النتائج:

- **البنية الوحيدة:** البنية الوحيدة في Mininet والموضحة في الشكل(4) تعني أنه لدينا مبدل واحد في الشبكة، ومجموعات المضيفين الموجودة في الشبكة تتصل جميعها بهذا المبدل، يتصل هذا المبدل بدوره بمتحكم SDN مركزي موجود في طبقة أعلى، وقد افترضنا أن هذه الشبكة تحتوي على 16 مضيف، ويتم إنشاؤها بالأمر التالي:

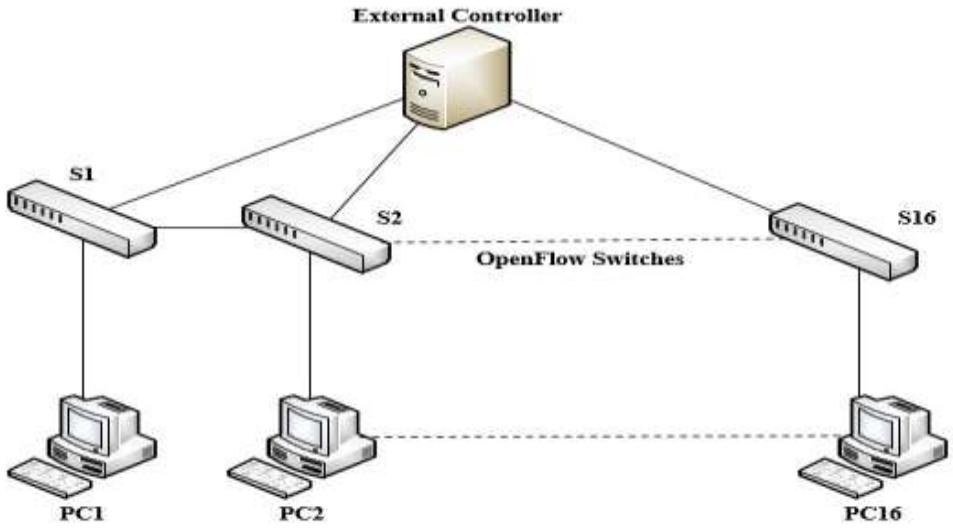
''$ sudo mn --topo = single,16 --controller remote,ip = 127.0.0.1 --mac''



الشكل (4) البنية الوحيدة

- **البنية الخطية:** البنية الخطية في Mininet الموضحة في الشكل(5) تعني أنه لدينا اتصال خطي بين المضيفين والمبدلات، حيث أن كل مضيف يتصل مع مبدله الخاص، والمبدلات تتصل فيما بينها بشكل خطي، وجميع هذه المبدلات تتصل بمتحكم مركزي موجود في طبقة أعلى، وقد افترضنا أن هذه الشبكة تحتوي على 16 مضيف، ويتم إنشاؤها بالأمر التالي:

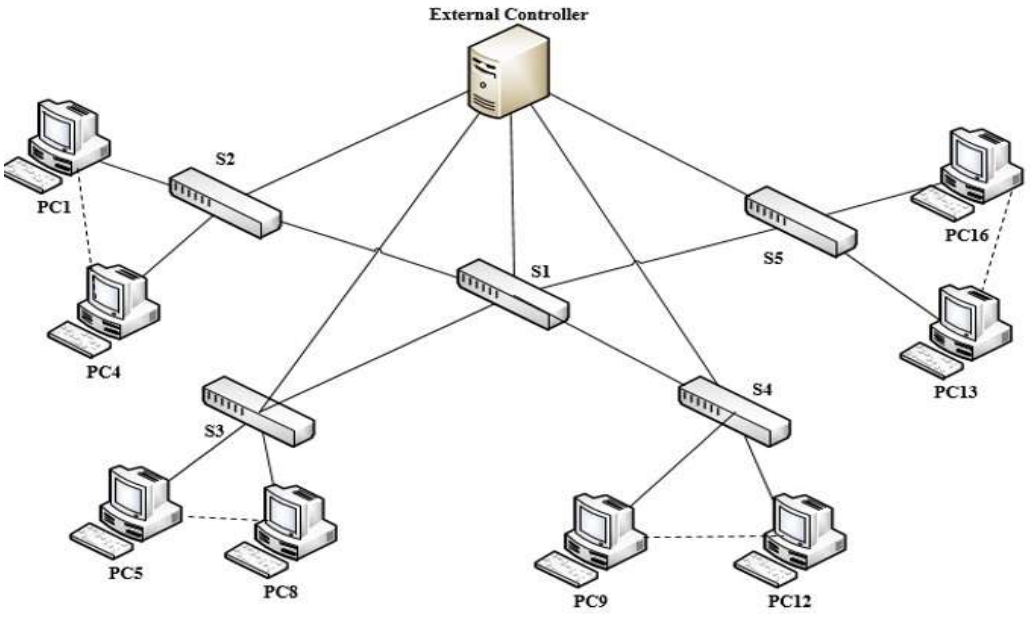
''$ sudo mn --topo = linear,16 --controller remote,ip = 127.0.0.1 --mac''



الشكل (5) البنية الخطية

➢ **البنية الشجرية:** البنية الشجرية في Mininet والموضحة في الشكل(6) تعني أن جميع المضيفين والمبدلات متصلة مع بعضها بشكل هرمي، وإن جميع هذه المبدلات تتصل بمتحكم SDN مركزي، وقد افترضنا أن هذه الشبكة تحتوي على 16 مضيف، ويتم إنشاؤها بالأمر التالي:

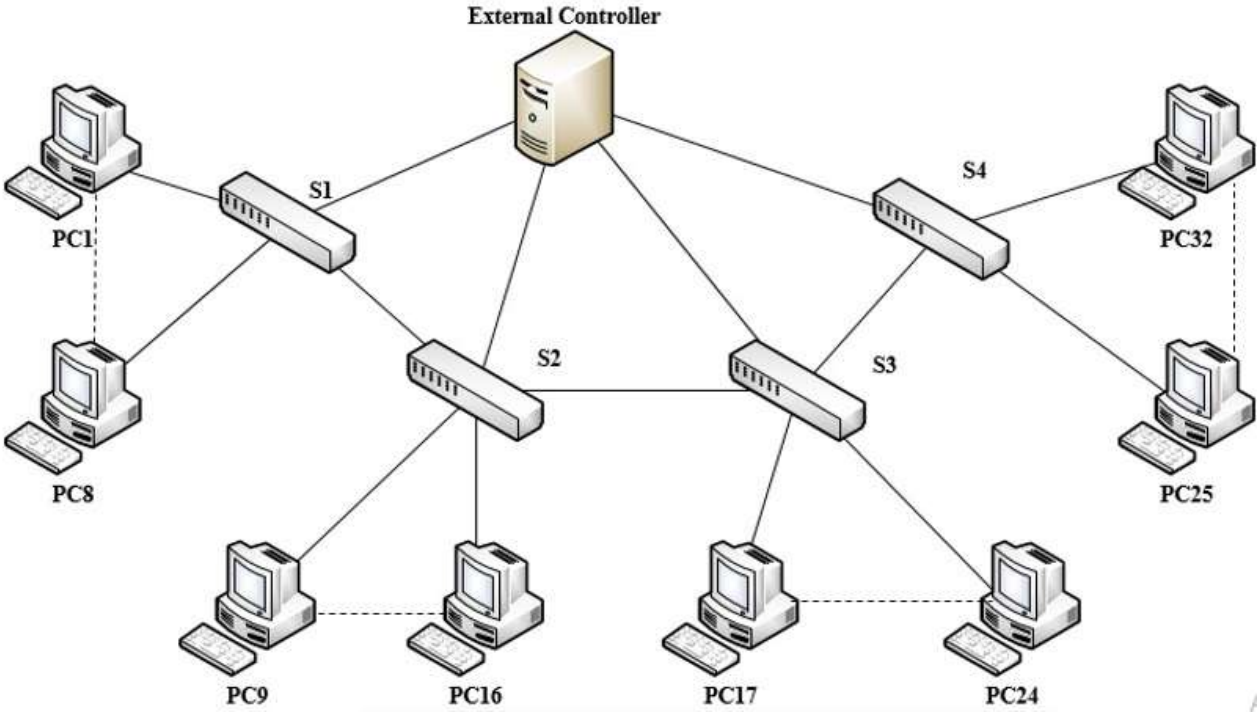
''$ sudo mn --topo = tree,depth = 2,fanout = 4 --controller remote,ip = 127.0.0.1--mac''



الشكل (6) البنية الشجرية

➢ **البنية المقترحة:** في هذه البنية اقترحنا وجود 4 مبدلات و 32 مضيف، حيث صممت الشبكة بحيث أن كل مبدل يملك نفس عدد المضيفين المتصل به (مثلاً 8 مضيف لكل مبدل)، وتكون المبدلات متصلة مع بعضها، وبدورها فإن جميع هذه المبدلات تتصل بمتحكم SDN مركزي، ويتم إنشاؤها بالأمر التالي:

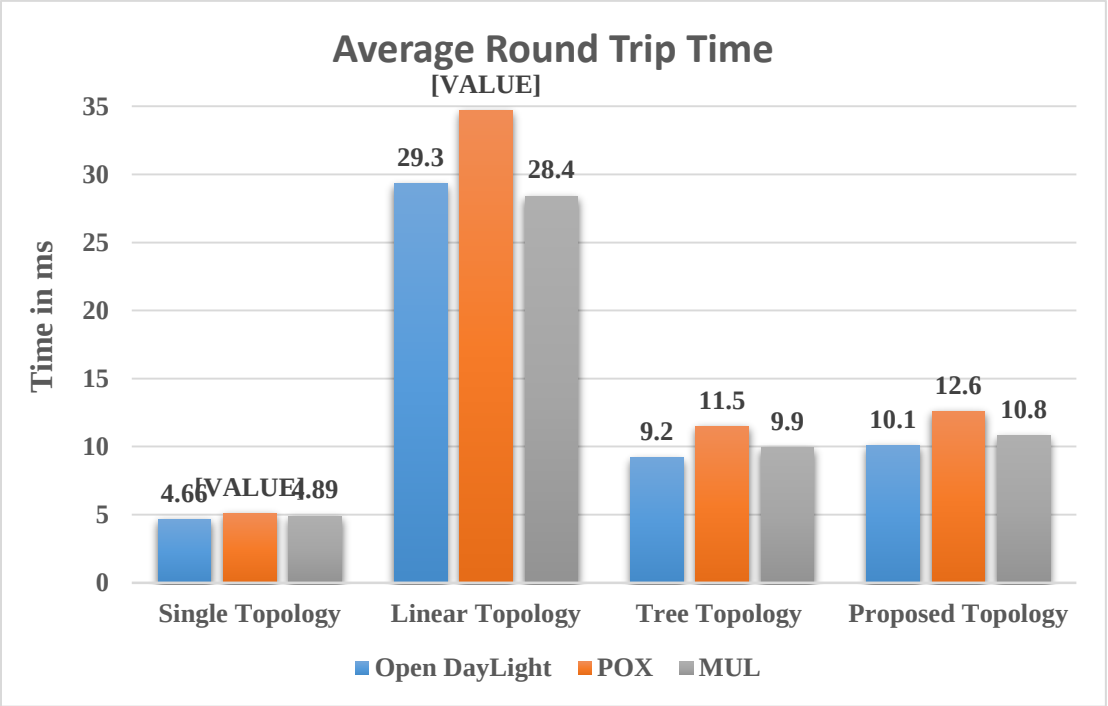
''sudo mn --custom 32hosts-4switches.py --topo mytopo --controller remote,ip = 127.0.0.1 --mac''



الشكل (7) البنية المقترحة

2-1- السيناريو الأول: زمن الرحلة في البنى الأربعة:

يمكن حساب زمن الرحلة من خلال استعلام رسالة ICMP باستخدام الأمر Ping أي عن طريق قياس الزمن بين الرسالة Echo Request والرسالة Echo Reply، ويعتمد أداء الشبكة ككل من خلال حساب المتوسط الحسابي للرسائل المرسلة عبر الشبكة ككل، حيث نقوم باختبار الأمر Ping لكل مضيف مع باقي الأجهزة ونقوم بحساب المتوسط الحسابي، وتظهر النتائج في الشكل (8):

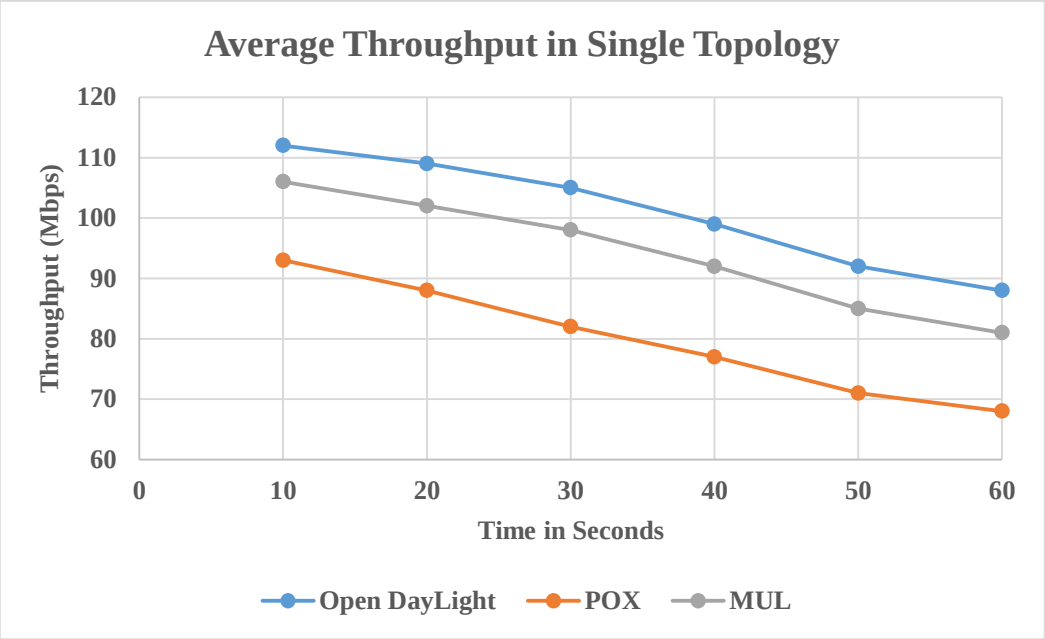


الشكل (8) متوسط زمن الرحلة في البيئات الأربعة

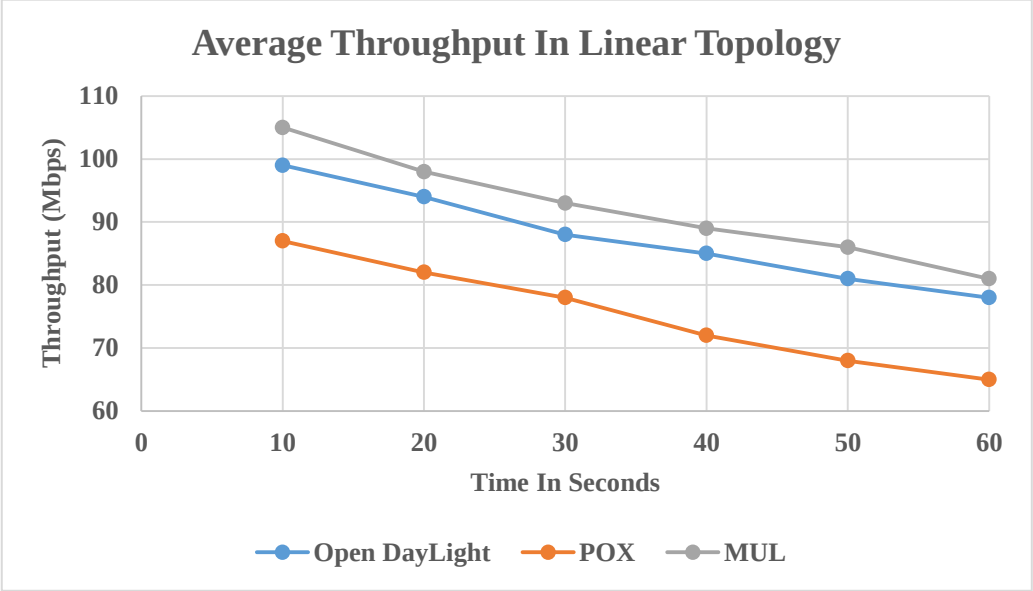
بالنظر إلى الشكل نرى أن المتحكم MUL المبرمج بلغة C يعطي استجابة أسرع في البنية الخطية، وذلك لأن لغة C تعتبر الأفضل في عمليات المعالجة عند وجود نيسب وحيد، ويليه المتحكم OpenDayLight المبرمج بلغة Java، ومن ثم المتحكم POX المبرمج بلغة Python. أما في البيئات الثلاث المتبقية وهي الوحيدة والشجرية والمقترحة نجد أن متحكم OpenDayLight يعطي زمن استجابة أسرع من المتحكم MUL، وذلك لأن لغة Java تدعم تعدد الأنماط بشكل فعال أكثر من لغة C، وفي البنى الأربع يعطي المتحكم POX أسوأ زمن استجابة بين المتحكمات المذكورة وذلك لأن لغة Python لاتدعم تعدد الأنماط أو تعدد النياسب بشكل جيد.

2-2- السيناريو الثاني: الإنتاجية في البيئات الأربعة:

الإنتاجية هي كمية البيانات التي تم تسليمها خلال وحدة معينة من الزمن. تم حسابها في كل من البيئات الأربع و توضيح النتائج بالأشكال 9-10-11-12:

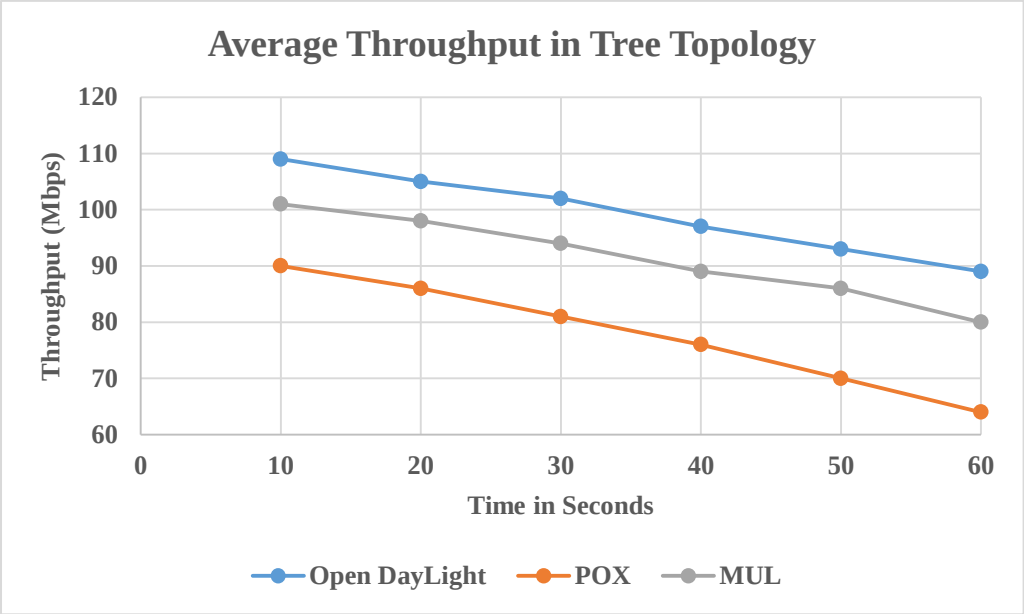


الشكل (9) متوسط الانتاجية في البنية الوحيدة

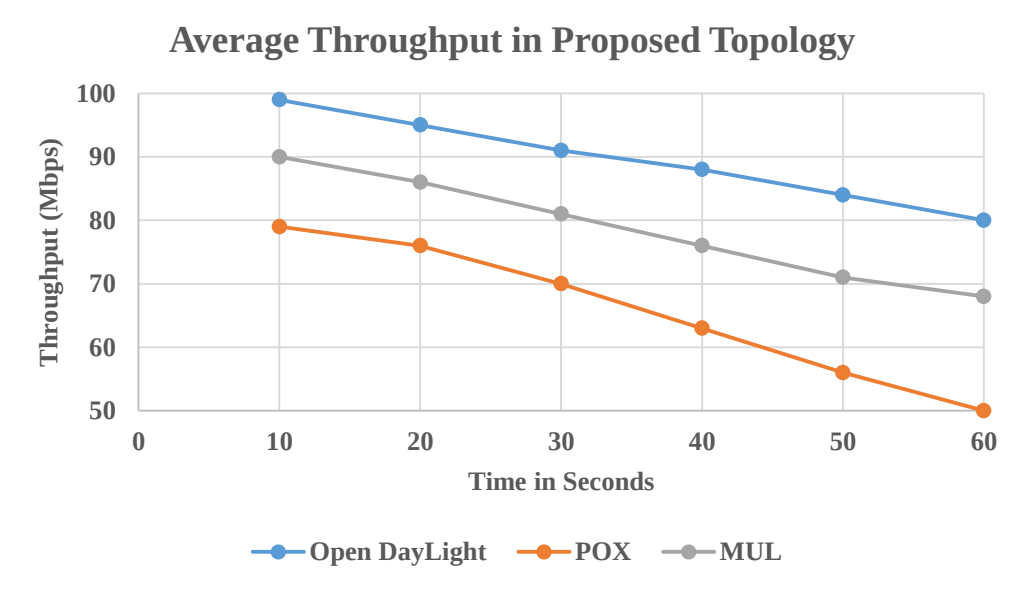


الشكل (10) متوسط الانتاجية في البنية الخطية

بالنظر إلى الشكل (10) نجد أنه مختلف عن الأشكال 9-11-12، حيث أن المتحكم MUL المبرمج بلغة C يعطي إنتاجية أعلى في البنية الخطية، وذلك لأنه كما ذكرنا سابقاً فإن لغة C تعتبر الأفضل في عمليات المعالجة عند وجود نيسب وحيد، ويليه المتحكم OpenDayLight المبرمج بلغة Java ومن ثم المتحكم POX المبرمج بلغة Python.



الشكل (11) متوسط الانتاجية في البنية الشجرية



الشكل (12) متوسط الانتاجية في البنية المقترحة

بالنظر إلى الأشكال 9-10-11-12 نستنتج أن الشبكات التي تملك متحكم OpenDayLight تعطي إنتاجية أعلى من الشبكات التي تحوي متحكم POX في جميع البنى المدروسة، وإنتاجية أعلى من الشبكات التي تحوي متحكم MUL في ثلاث من البنى الأربع المدروسة، وبما أننا نبحث عن متحكم يعطي أفضل أداء في بنية ليست خطية فإننا سوف نختار المتحكم OpenDayLight لاستخدامه في الدراسات المستقبلية التي نحتاجها، ولا سيما أنه يدعم عدداً كبيراً من التطبيقات ويحوي عدداً كبيراً من الميزات غير الموجودة في متحكمات أخرى.

الاستنتاجات والتوصيات:

قمنا بمقارنة ثلاث متحكمات رئيسية هي متحكم POX المبرمج بلغة Python، ومتحكم MUL المبرمج بلغة C، ومتحكم OpenDayLight المبرمج بلغة JAVA في أربعة بنى مختلفة من شبكات SDN وهي البنية الوحيدة والخطية والشجرية وبنية مقترحة، تمت عمليات مقارنة الأداء من خلال قياس عاملين هما: الإنتاجية Throughput وزمن الرحلة Round Trip Time. ومن خلال التجربة تبين أن متحكم OpenDayLight يعطي أداءاً أفضل من متحكم POX في جميع البنى المدروسة، و أداءاً أفضل من متحكم MUL في ثلاث من البنى الأربع المدروسة وفق العاملين الذين قمنا بدراستهما وفي البيئات المتنوعة التي تمت الدراسة فيها.

من جهة أخرى فإن المتحكم OpenDayLight يملك واجهة رسومية يمكن الوصول إليها بسهولة للحصول على معلومات عن البنية التحتية للشبكة المستخدمة، كما أنه يدعم عدداً كبيراً من التطبيقات والمزايا مما يسمح باستخدامه في مجالات بحثية أوسع من باقي المتحكمات.

References:

[1] AHMAD SAKER AHMAD; AFRAA MOHAMMAD, *Improving Bandwidth Utilization in Software Defined Networks (SDN)*, Tishreen University Journal for Research and Scientific Studies-Engineering Sciences Series Vol. (64) No. (5) 4242.

[2] DMITRY DRUTSKOY; ERIC KELLER, and JENNIFER REXFORD, *Scalable Network Virtualization in Software Defined Networks*, IEEE Internet Computing 2013.

[3] PING SONG; YI LIU; TIANXIAO LIU, and DEPEI QIAN, *Controller-Proxy: Scaling Network Management for Large-Scale SDN Networks*, Computer Communications 2017, DOI:10.1016/j.comcom.2017.05.002

[4] GENG LIANG, and WEN LI, *A novel industrial control architecture based on Software Defined Network*, Measurement and Control 2018, Vol. 51(7-8) 360 –367.

[5] MOHAMMAD ASHRAFI; NOELIA CORREIA , and FAROQ AL-TAM, *A Scalable and Reliable Model for the Placement of Controllers in SDN Networks*, Springer Nature Switzerland AG 2019, https://doi.org/10.1007/978-3-030-05195-2_8

[6] M. C. Dacier; H. Konig, R. Cwalinski; F. Kargl, and S. Dietrich, *Security Challenges and Opportunities of Software-Defined Networking*, IEEE Security Privacy 2019, vol. 15, no. 2, pp. 96.

[7] KURDMAN ABDULRAHMAN RASOL RASOL; JORDI DOMINGO-PASCUAL, *Joint Latency and Reliability Aware Controller Placement*, International Conference on Information Networking (ICOIN),2021, DOI: 10.1109/ICOIN50884.2021.9333864 .

[8] ALIREZA SHIRMARZ; ALI GHAFFARI, *Performance issues and solutions in SDN-based data center: a survey*, The Journal of Supercomputing: Springer Science + Business Media, LLC, 2020, https://doi.org/10.1007/s11227-020-03180-7

[9] KONSTANTINOS POLULARAKIS; GEORGE IOSIFIDIS; GEORGIOS SMARAGDAKIS, and LEANDROS TASSIULAS, *One Step at a Time: Optimizing SDN Upgrades in ISP Networks*, IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, doi:10.1109/infocom.2017.8057136

[10] JACOB H. COX; JOAQUIN CHUNG; SEAN DONOVAN; JARED IVEY; RUSSELL J.CLARK; GEORGE RILEY, AND HENRY L. OWEN, *Advancing Software-Defined Networks:A Survey*, IEEE Access 2017, doi:10.1109/ACCESS.2017.2762291

[11] RASHID AMIN; MARTIN REISSLEIN, and NADIR SHAH; *Hybrid SDN Networks: A Survey of Existing Approaches*, IEEE Communications Surveys & Tutorials 2018, DOI:10.1109/COMST.2018.2837161
[12] FETIA BANNOUR; SAMI SOUIHI, and ABDELHAMID MELLOUK, *Distributed SDN Control: Survey, Taxonomy and Challenges*, IEEE Communications Surveys & Tutorials 2017, doi:10.1109/comst.2017.2782482
[13] DAN LEVIN; ANDREAS WUNDSAM; BRANDON HELLER; NIKHIL HANDIGOL, and ANJA FELDMANN, *Logically Centralized? State Distribution Trade off sin Software Defined Networks*, HotSDN '12: Proceedings of the first workshop on Hot topics in software defined networks August 2012.
[14] LUYUAN FANG; FABIO CHIUSSI; DEEPAK BANSAL; VIJAY GILL; TONY LIN; JEFF COX, and GARY RATTERREE, *Hierarchical SDN for the hyper-scale, hyper-elastic data center and cloud*, The 1st ACM SIGCOMM Symposium 2015, DOI: 10.1145/2774993.2775009
[15] ERMIN SAKIE, and WOLFGANG KELLERER, *Impact of Adaptive Consistency on Distributed SDN Applications: An Empirical Study*, IEEE Journal on Selected Areas in Communications 2018, doi:10.1109/jsac.2018.2871309
[16] MOHAMMED ALSAEEDI; MOHD MURTADHA MOHAMAD, and ANAS A. AL-ROUBAIE, *Toward Adaptive and Scalable OpenFlow-SDN Flow Control: A Survey*, IEEE Access 2019, doi:10.1109/access.2019.2932422