

تحسين استهلاك عرض الحزمة في الشبكات المعرفة بالبرمجيات

د. أحمد صقر أحمد*

عفراء محمد**

(تاريخ الإيداع 24 / 2 / 2020. قَبِلَ للنشر في 13 / 9 / 2020)

□ ملخص □

تعد الشبكات المعرفة بالبرمجيات (SDN) Software Defined Networks النقلة النوعية في مجال الشبكات لكونها تفصل عناصر التحكم عن عناصر التوجيه، واقتصرت وظيفة عناصر التوجيه على تنفيذ القرارات التي ترسل إليها من المتحكم عن طريق بروتوكول التدفق المفتوح (OpenFlow Protocol (OF الذي يستخدم بشكل أساسي في SDN. نوضح في هذا البحث الاستفادة من المفهوم الجديد الذي قدمته الشبكات المعرفة بالبرمجيات، حيث تجعل إدارة الشبكة أسهل، فبدلاً من كتابة القواعد على كل جهاز، نقوم ببرمجة التطبيق الذي نريده في المتحكم، وتقوم أجهزة البنية التحتية بتنفيذ الأوامر الواردة إليها من المتحكم.

ولكي نحقق أفضل أداء لهذه التقنية لا بد من تطبيق جودة الخدمة (Quality of Service (QoS ضمنها، حيث تشمل جودة الخدمة عدة معايير أهمها عرض الحزمة المستخدم والتأخير وخسارة البيانات والرجعة (Jitter). ومن أهم هذه العوامل هو عرض الحزمة، لأنه من خلال تحسين هذا المعيار، يمكننا تحسين باقي المعايير الأخرى، لذلك نقدم في هذا البحث التحسين اللازم على المتحكم RYU لاستخدام عرض الحزمة بالشكل الأفضل، مما يحسن من جودة الخدمة في SDN.

الكلمات المفتاحية: الشبكات المعرفة بالبرمجيات، جودة الخدمة، عرض الحزمة، المتحكم RYU.

* أستاذ - قسم النظم والشبكات الحاسوبية - كلية الهندسة المعلوماتية - جامعة تشرين - اللاذقية - سورية.
E-mail: Ahmad.s.ahmad@tishreen.edu.sy

** طالبة دكتوراه - قسم النظم والشبكات الحاسوبية - كلية الهندسة المعلوماتية - جامعة تشرين - اللاذقية - سورية.
E-mail: afraarasheedmohammad@tishreen.edu.sy

Improving BandWidth Utilization in Software Defined Networks (SDN)

Dr. Ahmad Saker Ahmad*
Afraa Mohammad**

(Received 24 / 2 / 2020. Accepted 13 / 9 / 2020)

□ ABSTRACT □

Software Defined Networks (SDN) is the qualitative movement in the field of networks due to that fact that it separates the control elements from the routing elements, and the function of the routing elements was limited to the implementation of the decisions that are sent to it by the controller through the OpenFlow Protocol (OF) which is mainly used in SDN.

We explain in this paper the benefit of the new concept which is presented by SDN and it makes network management easier, so instead of writing the rules on each device, we program the application in the controller, and the infrastructure devices run the received commands from the controller.

In order to achieve the best performance of this technology, a Quality of Service (QoS) must be applied within it, where it includes several criteria, the most important are the used bandwidth, delay, packet loss and jitter.

The most important of these criteria is the bandwidth, because by improving this standard, we can improve the rest of the other criteria.

Therefore, in this paper, we provide the necessary improvement on the RYU controller to use the best bandwidth, which improves the quality of service in SDN.

Keywords: Software Defined Networks, Quality of Service, BandWidth ,RYU Controller.

* Professor, Department of Computer Networks & System, Faculty of Information Technology, University of Tishreen, Lattakia, Syria. E-mail: Ahmad.s.ahmad@tishreen.edu.sy

** PhD Student, Department of Computer Networks & System, Faculty of Information Technology, University of Tishreen, Lattakia, Syria. E-mail: afraarasheedmohammad@tishreen.edu.sy

مقدمة:

تعد الشبكات المعرفة بالبرمجيات (SDN) Software Defined Networks من أهم التقنيات الحديثة في مجال الشبكات، لما تقدمه من مزايا في معمارية وإدارة الشبكات والتحكم بكامل وظائفها. تتميز SDN عن الشبكات التقليدية بوجود عنصر التحكم المركزي وهو المتحكم الذي يكون مسؤولاً عن كل العمليات التي تحدث في الشبكة.

يتوضع المتحكم في طبقة التحكم التي تمثل الطبقة الثانية من معمارية SDN، ويرسل الأوامر إلى طبقة البنية التحتية من خلال بروتوكول التدفق المفتوح (OpenFlow (OF)، وهذا يوضح كيف أن SDN قامت بعزل المستوى البرمجي عن مستوى العتاديات، مما يسهل إدارة الشبكة ويحسن من أدائها[1].

يتميز المتحكم بأنه يمتلك نظرة كاملة عن كل طوبولوجيا الشبكة، ويتخذ القرارات اللازمة لتوجيه الرزم، ومن خلاله يمكن برمجة عدة تطبيقات مختلفة ومن ثم إرسال التعليمات إلى الأجهزة في طبقة البنية التحتية. لهذا المتحكم عدة أنواع منها [2]: POX –RYU- NOX-FloodLight-ONOS-OpenDayLight-Beacon. ولكي نحدد مدى فعالية أي تقنية جديدة لا بد من تقييم أدائها وفق معايير مختلفة، من أهمها جودة الخدمة Quality of Service(QoS) التي توضح كيفية التعامل مع أنواع مختلفة من الرزم وفقاً لحساسيتها لعدة عوامل منها التأخير (Delay) والرجرجة (Jitter) وعرض الحزمة (BandWidth) وخسارة الرزم (Packet Loss). حيث تختلف التطبيقات عن بعضها البعض بمتطلبات تحقيق جودة الخدمة [3].

كما تؤكد الدراسات أن الشبكات المعرفة بالبرمجيات هي بحد ذاتها تطوير من أجل تحقيق QoS [4]، فبدلاً من تنفيذ الخطوات اللازمة لتحقيق QoS في كل جهاز من الأجهزة الموجودة في الشبكات التقليدية، نكتفي ببرمجة المتحكم لكي يتم تطبيق جودة الخدمة في كامل الشبكة.

أهمية البحث وأهدافه:

تبرز أهمية البحث في أن الشبكات المعرفة بالبرمجيات تعد مستقبل الشبكات ولا بد من أن تقدم أفضل أداء، وخاصة أن الأبحاث التي تتناول QoS في SDN لا تزال قليلة، مما يوضح ضرورة البحث في هذا المجال. يتناول هذا البحث مفهوم تقنية SDN، وطرق تحقيق جودة الخدمة QoS، ويهدف إلى تقديم آلية لتحسين استخدام عرض الحزمة بالشكل الأمثل في SDN باستخدام المتحكم RYU، حيث أن عرض الحزمة يلعب دوراً أساسياً في تقييم أداء الشبكة، ويعود تأثيره على كل المعايير الأخرى المتعلقة بجودة الخدمة.

طرائق البحث ومواده:

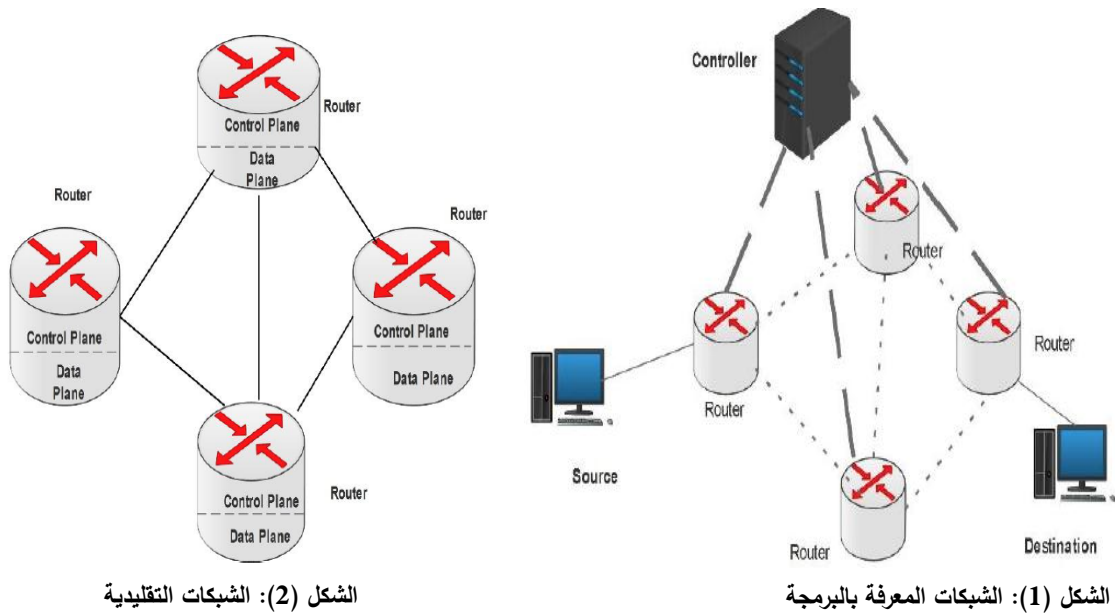
اعتمدنا في البحث على المحاكى Mininet وهو محاكي مفتوح المصدر، ويستخدم من أجل بناء شبكات SDN [5]، إضافةً إلى الأداة Miniedit لرسم طوبولوجيا الشبكة. واعتمدنا من أجل قياس عرض الحزمة على الأداة Iperf، التي تقوم بإنشاء تطبيق مخدم/زبون (Client/Server) وترسل الرزم بينهما سواء رزم بروتوكول التحكم بالنقل Transmission Control Protocol(TCP) أو رزم

بروتوكول حزم بيانات المستخدم (User Datagram Protocol (UDP) ، ومن ثم تقيس عرض الحزمة وفق فاصل زمني يحدد من قبل المستخدم [6] .

1- الشبكات المعرفة بالبرمجيات (SDN): Software Defined Networks

قدمت الشبكات المعرفة بالبرمجيات تطوراً كبيراً في معمارية الشبكات بما يتناسب مع متطلبات التطوير المستمر في عالم الشبكات [7].

تقوم تقنية SDN على فصل عنصر التحكم عن باقي الأجهزة كما يوضح الشكل (1)، وهذا ما يميزها عن الشبكات التقليدية التي يكون فيها عنصر التحكم مضمناً ضمن عنصر التوجيه كما يوضح الشكل (2).



لذا تمثل SDN طريقة جديدة في إعداد الشبكة وإدارتها وتحديثها بمرونة وكفاءة عالية، على العكس من الشبكات التقليدية، مما يشجع الانتقال نحو استخدامها [8].

من أهم مكونات هذه التقنية الجديدة هو المتحكم الذي يتواصل مع باقي الأجهزة من خلال بروتوكول التدفق المفتوح (OpenFlow Protocol (OF)) الذي يعمل وفق TCP ويقوم بتخزين قواعد التوجيه ضمن جداول التدفق بحيث يتكون كل مدخل من هذه المداخل من مجموعة قواعد توضح إما توجيه الرزم أو إهمالها تبعاً لما يقرره المتحكم. يمتلك المتحكم نظرة شاملة عن كل طوبولوجيا الشبكة وذلك لأن البروتوكول OF يقوم بكشف الوصلات في الشبكة من خلال بروتوكول كشف حالة الارتباط (LLDP) Link Layer Discovery Protocol [9].

2- المتحكم RYU:

يمثل المتحكم المكون الأهم في SDN، ومن خلاله يتم برمجة التطبيقات المختلفة للشبكة. تختلف المتحكمات عن بعضها البعض بلغة البرمجة التي تدعمها، ونسخة البروتوكول الذي تعمل وفقه، ودعم تعدد النياص، إضافةً إلى اختلافات أخرى تتعلق بمجال استخدام هذه المتحكمات سواء في مراكز البيانات والحوسبة السحابية وغير ذلك.

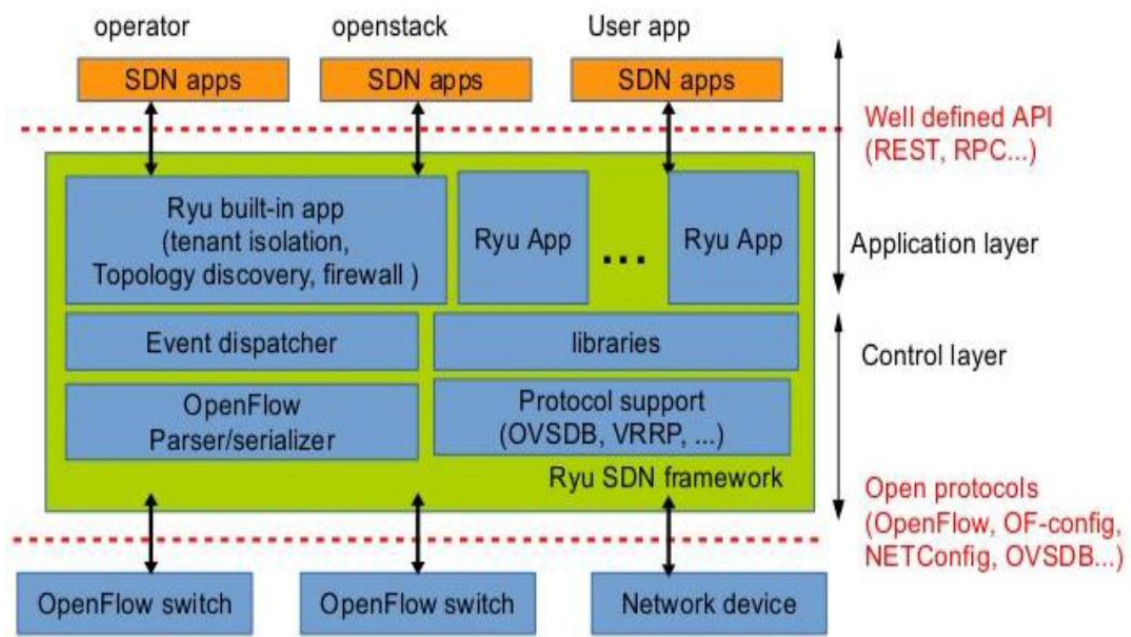
يوضح الجدول (1) مقارنة بين المتحكمات (RYU, Floodlight, OpenDayLight, ONOS, POX) من حيث لغة البرمجة، ونسخة البروتوكول التي يدعمها كل متحكم، ودعم تعدد النياسب، و دعم نظام التشغيل، وفيما إذا كان المتحكم مفتوح المصدر مما يمكن تطويره أو لا [10].

الجدول (1): مقارنة بين المتحكمات

المتحكم	لغة البرمجة	نسخة البروتوكول OF التي يدعمها	دعم تعدد النياسب	دعم نظام التشغيل	مفتوح المصدر
RYU	Python	OF 1.0 , 1.1 , 1.2 , 1.3 , 1.4 , 1.5	نعم	Linux	نعم
Floodlight	JAVA	OF 1.0 , 1.1 , 1.2 , 1.3 , 1.4 , 1.5	نعم	Linux, Windows,MAC	نعم
OpenDayLight	JAVA	OF 1.0 , 1.1 , 1.2 , 1.3 , 1.4 , 1.5	نعم	Linux, Windows,MAC	نعم
ONOS	JAVA	OF 1.0 , 1.1 , 1.2 , 1.3 , 1.4 , 1.5	نعم	Linux, Windows,MAC	نعم
POX	Python	OF 1.0	لا	Linux, Windows,MAC	نعم

ومن أشهر هذه المتحكمات هو المتحكم RYU المفتوح المصدر، وتمت برمجته بلغة بايثون، ويدعم حتى الإصدار OF1.5 من بروتوكول التدفق المفتوح .

يوضح الشكل (3) مكونات المتحكم RYU التي تسهل تطوير التطبيقات الشبكية، مما يسهل إدارة الشبكة ومنها (OpenFlow-Config(OFconfig التي تستخدم من أجل إعدادات البروتوكول OF، والمكتبة (Open Virtual Switch DataBase (OVSDB وتستخدم من أجل إعدادات المبدل لكي يتعامل مع البروتوكول OF، بما يشمل إنشاء وحذف وتعديل القواعد في جدول التدفق. والمكتبة (Network Configuration (NETConfig التي تساعد في تنفيذ الإعدادات على الأجهزة الموجودة في الشبكة. المكون المسؤول عن تنفيذ التطبيقات في المتحكم RYU وجعلها قابلة للتشغيل هو ryu-manager. ومن هذه التطبيقات، تطبيق اكتشاف طوبولوجيا الشبكة (Topology Discovery)، وتطبيق الجدار الناري (Firewall)، والتطبيق (Switching -Hub) ليقوم المتحكم بدور Hub، وغير ذلك من التطبيقات المضمنة، والتطبيقات الأخرى التي يمكن برمجتها وتنفيذها ضمن المتحكم [11].



الشكل (3): بنية المتحكم RYU

3- جودة الخدمة QoS في الشبكات:

تعد جودة الخدمة من أهم المعايير التي تميز أداء الشبكات عن بعضها البعض، وتمثل مجموعة الأدوات التي تقوم بتطبيق معالجة مختلفة للبيانات وفقاً لحساسية هذه البيانات لعدة عوامل أهمها [12]:

1. عرض الحزمة (BandWidth (BW): يمثل سعة الوصلة، أي عدد البتات التي يمكن إرسالها ضمن هذه الوصلة في الثانية الواحدة، وتقاس بوحدة بت/ثانية (bps) bit per second.
2. التأخير (Delay): يمثل التأخير الذي يحدث في الشبكة أثناء نقل الرزم، ويقاس بوحدة ثانية (sec) second.
3. الرجرجة (Jitter): يمثل الإزاحة الزمنية بين الرزم، أي الانحراف المعياري في أزمته وصول الرزم ويقاس بوحدة ميلي ثانية (ms) millisecond.

مثلاً: في حال وصلت أول رزمة بتأخير 3sec ، ووصلت الرزمة الثانية بتأخير 3sec ، ووصلت الرزمة الثالثة بتأخير 3sec، في هذه الحالة لا يوجد Jitter لأن الفواصل الزمنية بين وصول الرزم هي نفسها. أما في حال وصلت الرزمة الأولى بتأخير 3sec، ووصلت الرزمة الثانية بتأخير 50sec، ووصلت الرزمة الثالثة بتأخير 100sec، عندها يوجد حالة عدم استقرار Jitter لأن الفواصل الزمنية بين وصول الرزم مختلفة.

4. خسارة الرزم (Packet Loss): تمثل مقدار الرسائل المفقودة، وتقاس عادة كنسبة مئوية من الرزم المرسله. تختلف التطبيقات في حاجاتها لكل من العوامل السابقة، فمثلاً يحتاج الفيديو بجميع أشكاله إلى مقدار كبير من عرض الحزمة، وقيمة منخفضة لـ Jitter ، بينما تطبيقات أخرى مثل البريد الإلكتروني تكون أقل حساسية لكل من العوامل الأربعة السابقة، ومن هذه الاختلافات تبرز أهمية تطبيق جودة الخدمة في الشبكات.

4- تقنيات تحقيق جودة الخدمة في الشبكات:

توجد عدة طرق تؤمن تطبيق جودة الخدمة في الشبكات نذكر منها [13]:

1. الخدمات المدمجة ((Integrated Services(IntServ)): وفيها يتم حجز الموارد مسبقاً لتدفق محدد، وتبقى هذه الموارد مخصصة لهذا التدفق على طول المسار دون الاهتمام بنوع الرزم، وتستخدم عادة بروتوكول حجز الموارد .Resource Reservation Protocol (RSVP).
 2. الخدمات التفاضلية ((Differentiated Services(DiffServ)): وتقوم على تحقيق جودة الخدمة المعتمدة على الصنف.
تصنف التدفقات إلى عدة أصناف وفقاً لما تحتاجه التدفقات من عرض الحزمة والتأخير وغير ذلك، وتعطي أولوية لأصناف دون الأخرى. أي أنها تقدم خدمة أفضل لبعض الأصناف من الخدمة التي تقدمها لأصناف أخرى، فمثلاً تعطي تطبيقات الوسائط المتعددة أولوية عالية عن باقي التطبيقات.
في هذا النوع من الخدمات لا نحتاج إلى حجز مسبق للموارد.
 3. إعطاء حركة النقل نمطاً معيناً (Traffic Shaping): وتقوم على تنظيم المعدل الوسطي لاستخدام عرض الحزمة المتاحة في نقل الرزم.
 4. الأرتال (Queues): تنظيم الرزم ضمن أرتال وإعطاء كل رتل ميزات وفقاً لنوع البيانات، مما يخفف الازدحام في الشبكة.
 5. التوافق على مستوى الخدمة (Service Level Agreement(SLA): وتتضمن الاتفاق بين طرفي الإرسال على المحددات الخاصة لتحقيق جودة الخدمة المطلوبة.
- 5- تقنيات تحقيق جودة الخدمة في شبكات SDN:**
- تعد تقنية SDN بحد ذاتها آلية مطورة لتحقيق QoS وذلك لكون المتحكم يتمتع برؤية شاملة عن كامل الشبكة، وبسبب الفصل الذي قدمته بين طبقتي التحكم والأجهزة.
- إلا أنه توجد عدة محاولات لتحقيق QoS في SDN مستمدة من التقنيات المستخدمة في الشبكات التقليدية ومنها [14]
1. تقنية FlowQoS: تقوم على حجز الموارد مسبقاً لأي تدفق، إلا أنها لم تتجح في الشبكات الكبيرة، ولذلك لم يطرأ عليها أي تحسينات.
 2. تقنية OpenQoS : وهو متحكم جديد يدعم QoS من خلال إعطاء الأولويات للتدفقات على بعضها الآخر، وصمم بشكل خاص من أجل تطبيقات الوسائط المتعددة، حيث يعطيها الأولوية الأعلى.
 3. تقنية QoSFlow: تركز على إدارة الأرتال وجدولة الرزم بحيث تضمن وصول الرزم إلى الهدف بأفضل مسار.
- نوضح في هذا البحث كيف يتم تحسين استهلاك عرض الحزمة في الشبكة وذلك باستخدام المتحكم RYU ، من خلال ضمان Traffic Shaping على الرزم، لذا لا يحدث حجز لكامل عرض الحزمة المتوفر من أجل تدفق محدد، وإنما حسب الحاجة الفعلية لعرض الحزمة.

النتائج والمناقشة:

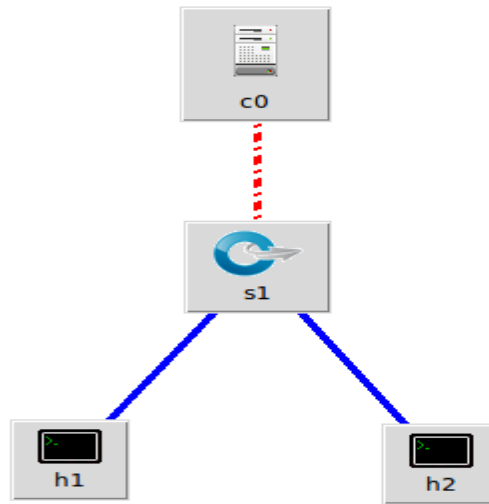
تم تنفيذ سيناريوهات لمقارنة استهلاك عرض الحزمة في حال استخدام منفذين مختلفين للاتصال هما المنفذ 5201 والثاني 5202، وذلك عند تشغيل نفس التطبيق باستخدام المتحكم RYU. من خلال مقارنة نتائج السيناريوهين الأول والثاني، وجدنا أنه لا بد من إيجاد طريقة فعالة لاستهلاك عرض الحزمة، وتم

تنفيذ هذه الطريقة وفق السيناريوهين الثالث والرابع، باستخدام منفذي الاتصال 5201 و 5202 والتطبيق نفسه، ومقارنتها مع نتائج السيناريوهين الأول والثاني.

إن الهدف من تغيير المنافذ (أرقام هذه المنافذ ليست خاصة بأي خدمة في SDN، ويمكننا استخدام منافذ أخرى مثل المنفذ 5001 والمنفذ 5002) هو معرفة مقدار عرض الحزمة المستهلكة وفق كل منفذ، ومن ثم سنخصص لكل منفذ قيم معينة لعرض الحزمة لكي يتم توجيه الرزم إلى المنفذ المناسب وفقاً لحجمها ومتطلباتها من عرض الحزمة. أظهرت النتائج فعالية هذه الطريقة في استهلاك عرض الحزمة بشكل ملائم لحجم الرزم المختلف التي يتم إرسالها في الشبكة.

9-1 السيناريو الأول:

اعتمدنا في هذه السيناريو على المحاكى Mininet، وقمنا باستخدام الأداة Miniedit برسم طوبولوجيا مؤلفة من متحكم c0 ومبدل ovs1 ومضيفين (h1, h2) كما يوضح الشكل (4).



الشكل (4): طوبولوجيا الشبكة

لكي نقوم بتشغيل الشبكة السابقة ونتأكد من أن الأجهزة قادرة على الاتصال مع بعضها، يجب أولاً تشغيل المتحكم. نستخدم في هذا البحث المتحكم RYU الذي يحوي عدة تطبيقات منها التطبيق simple_switch_13 وفيه يستخدم المتحكم النسخة OF 1.3 من بروتوكول التدفق المفتوح، ومن خلال هذا التطبيق يقوم المتحكم بدور Switching Hub أي يتعلم عنوان الـ MAC للأجهزة المتصلة مع منفذ معين، ويحتفظ بها في جدول عناوين الـ MAC. وعندما تصل رزمة إلى جهاز ذو عنوان MAC مخزن في الجدول، يتم توجيهها إلى المنفذ المناسب، وإذا لم يكن عنوان MAC للجهاز معروفاً تحدث حالة غمر (flooding) في الشبكة.

نقوم بتشغيل المتحكم RYU لكي ينفذ التطبيق simple_switch_13. بعد تشغيل المتحكم RYU، تصبح الشبكة فعالة وتستطيع الأجهزة أن تتبادل الرزم فيما بينها، وبحيث أي تغير في الشبكة سيكون المتحكم على علم به، وذلك من خلال البروتوكول OF.

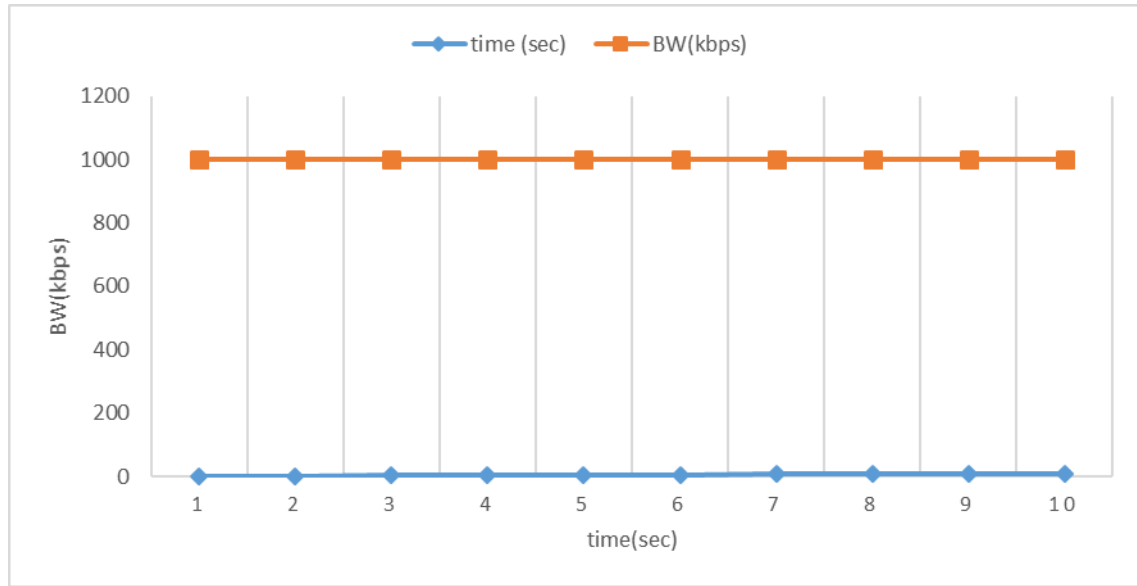
ثم نقوم بإنشاء تطبيق مخدم/زبون (client/server) باستخدام الأداة iperf بحيث يكون المضيف h1 هو المخدم (server)، وينقل بيانات من نوع UDP، ويتصل على المنفذ 5201، مع فاصل زمني لمدة ثانية واحدة.

أما المضيف الثاني h2 يقوم بدور زبون ويطلب الاتصال مع المخدم h1 ذو العنوان IP: 10.0.0.1 على المنفذ 5201 للحصول على بيانات من نوع UDP وعرض الحزمة الأعظمي هو 1Mbps. يوضح الجدول (2) السيناريو الأول.

الجدول (2): محددات السيناريو الأول

Scenario	Server	Client	Protocol	Port	Bandwidth	Interval
1 st . Scenario	H1 (IP:10.0.0.1)	H2 (IP:10.0.0.2)	UDP	5201	1 Mbps	1 sec

عند تنفيذ هذا التطبيق نجد أن قيمة عرض الحزمة المستهلكة هي بشكل تقريبي كامل عرض الحزمة 1Mbps كما يوضح الشكل (5). وبالتالي تم استهلاك كامل عرض الحزمة خلال زمن التجربة 10 sec.



الشكل (5): عرض الحزمة المستهلكة في السيناريو الأول

2-9 السيناريو الثاني:

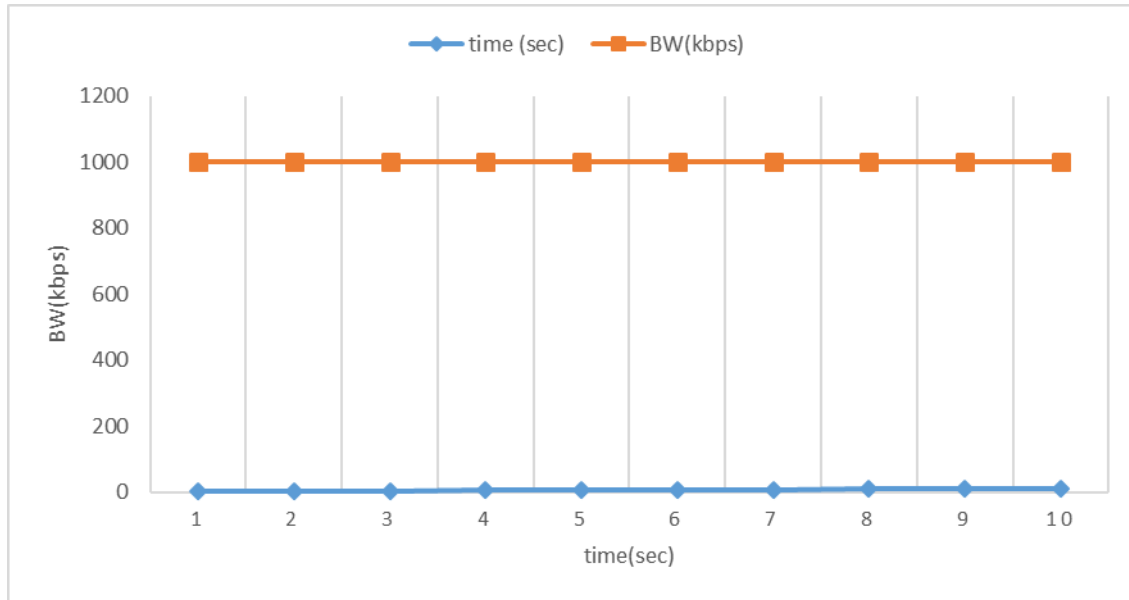
نستخدم الطوبولوجيا الموضحة في الشكل (4)، وننفذ التطبيق iperf وفق المحددات الموضحة في الجدول (3).

الجدول (3): محددات السيناريو الثاني

Scenario	Server	Client	Protocol	Port	Bandwidth	Interval
2 nd . Scenario	H1 (IP:10.0.0.1)	H2 (IP:10.0.0.2)	UDP	5202	1 Mbps	1 sec

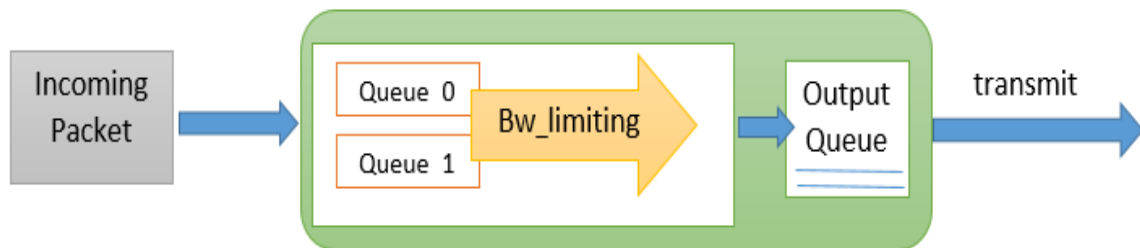
حيث في هذه السيناريو نشغل التطبيق simple_switch_13 ولكن هذه المرة على المنفذ 5202، بحيث h2 هو الزبون و h1 مخدم.

الهدف من تغيير المنفذ في هذه السيناريو هو معرفة فيما إذا كان تغيير منفذ الاتصال يؤثر على استهلاك عرض الحزمة عن السيناريو السابق التي استخدمنا فيه المنفذ 5201. توضح النتائج أن عرض الحزمة المستخدم أيضاً هو كامل عرض الحزمة كما يوضح الشكل (6).



الشكل (6): عرض الحزمة المستخدم في السيناريو الثاني

نلاحظ من نتائج تنفيذ السيناريو الأول والثاني أنه تم استخدام كامل عرض الحزمة، ومنه نستنتج أنه لا بد من تطبيق آلية تحقق أفضل استهلاك لعرض الحزمة بحيث لا يتم حجز كامل عرض الحزمة دون الحاجة لذلك. نقدم في هذا البحث آلية للحد من استهلاك عرض الحزمة بحيث لا يتجاوز عرض الحزمة مقداراً محدداً وفقاً للمنفذ المتصل به الجهاز، وعندها في حال كانت البيانات تحتاج عرض حزمة كبيرة يتم توجيهها في المسار الذي يستخدم عرض حزمة كبير، وإلا يتم توجيهها في مسار آخر. يوضح الشكل (7) الآلية المقترحة.

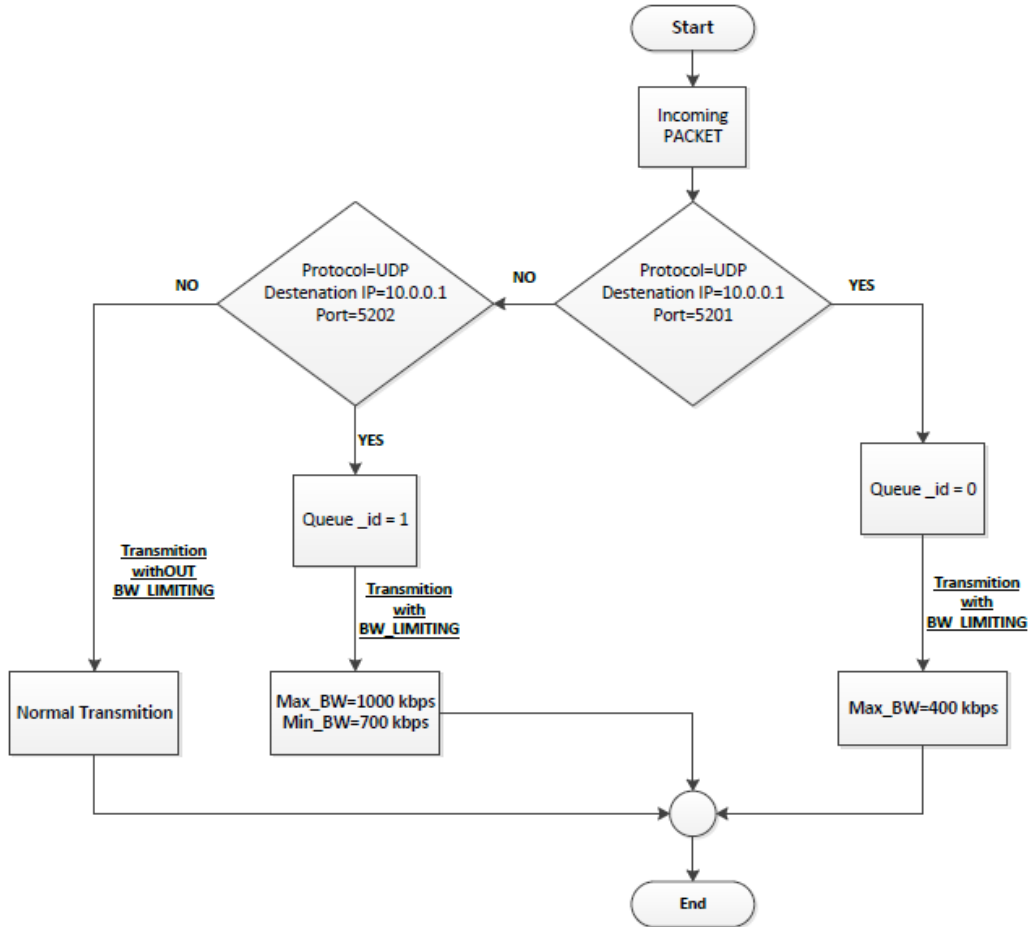


الشكل (7): الآلية المقترحة

تعتمد هذه الآلية على التدفق (per-flow) بحيث تقوم بحجز مسبق لعرض الحزمة على طول المسار أي تتطلب عملية إعداد مسبقة موضحة في الخطوات الآتية:

- 1- ننشئ رتلاً بمعرف `queue_id = 0` ، عرض الحزمة الأعظمي للرمز التي تمر عبره هو 400kbps .

- 2- ننشئ رتلاً بمعرف $queue_id = 1$ ، عرض الحزمة الأعظمي للرزم التي تمر عبره هو 1Mbps، وعرض الحزمة الأصغري هو 700kbps.
- اختبرنا القيم (400kbps, 1Mbps, 700kbps) بعد تنفيذ المقترح عدة مرات مع قيم مختلفة، لأنها أظهرت فعاليتها وذلك مع القيم المختلفة لحجم الرزم التي نريد نقلها في الشبكة.
- بعد إنشاء الرتل الذي يوضح كيفية توجيه الرزم ضمن أي عرض حزمة مناسب، يجب أن نوضح ماهي شروط وضع الرزم ضمن كل رتل لذلك لا بد من وضع قواعد لذلك وفق الآتي:
- 3- نضيف القواعد التالية إلى جدول التدفق الخاص بالمبدل sw1:
- Rule_1: إذا كان نوع الرزم UDP، ومرسلة إلى الجهاز ذو العنوان 10.0.0.1 عبر المنفذ 5201 توضع ضمن الرتل $queue_id = 0$.
- Rule_2: إذا كان نوع الرزم UDP، ومرسلة إلى الجهاز ذو العنوان 10.0.0.1 عبر المنفذ 5202 توضع ضمن الرتل $queue_id = 1$.
- وفي حال لم تطابق الرزم أي من هذه القواعد يتم نقلها بشكل عادي كما في السيناريو الأول والثاني، أي لا تطبق عليها الآلية المقترحة.
- يوضح الشكل (8) المخطط التدفقي (Flow Chart) للآلية المقترحة.



الشكل (8): المخطط التدفقي (Flow Chart) للآلية المقترحة

يتم تنفيذ الخطوات السابقة وإجراء التعديلات والتعليمات الجديدة اللازمة في المتحكم RYU لكونه مفتوح المصدر وبلغة بايثون مما يسهل برمجته وتنفيذ المقترح الجديد، وبالتالي من خلال تشغيل التطبيق الجديد ضمن المتحكم، فإن جميع العقد في الشبكة تلتزم بتنفيذه لكونها تتلقى الأوامر من المتحكم. إن المقترح الجديد يعمل على إعطاء عرض الحزمة نمطاً محدداً أي يجب أن يكون استخدام عرض الحزمة ضمن قيم محددة، بحيث تبقى الحزمة متاحة لتطبيقات أخرى. من خلال هذا المقترح يقوم المتحكم RYU بإعلام المبدل بهذه الآلية، وبالتالي عندما تصل أي رزمة إلى المبدل، يقوم بمطابقة القواعد المخزنة لديه مع هذه الرزم لكي يتم توجيهها في المسار المناسب.

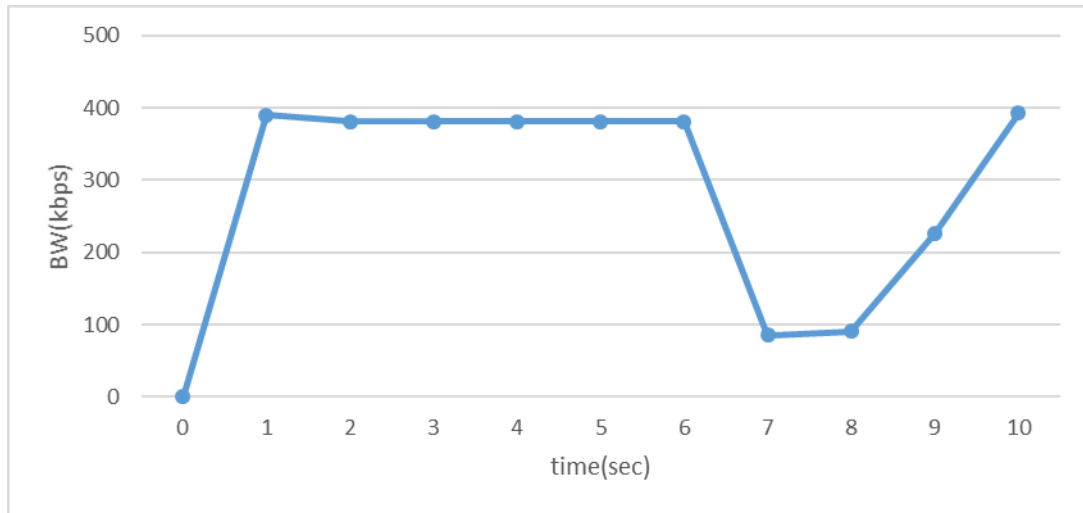
3-9 السيناريو الثالث:

نتحقق في هذه السيناريو من الآلية المقترحة حيث نستخدم الطوبولوجيا الموضحة في الشكل (4)، وننفذ التطبيق iperf وفق المحددات الموضحة في الجدول (4).

الجدول (4): محددات السيناريو الثالث

Scenario	Server	Client	Protocol	Port	Bandwidth	Interval
3 rd . Scenario	H1 (IP:10.0.0.1)	H2 (IP:10.0.0.2)	UDP	5201	1 Mbps	1 sec

يوضح الشكل (9) نتائج تنفيذ السيناريو الثالث.



الشكل (9): عرض الحزمة المستخدم في السيناريو الثالث

يظهر الشكل (9) أن جميع الرزم التي تطابق القاعدة الأولى (Rule_1)، لم تتجاوز عرض الحزمة فيها القيمة 400 kbps. حافظت قيمة BW خلال الفترة من 1sec إلى 6sec على قيم متقاربة من بعضها وتساوي تقريباً 381 kbps حيث كان حجم البيانات المرسله يساوي وسطياً 50 KBytes، أما عند الزمن 7 ثانية فقد بلغت 85.5 kbps وسبب هذا الانخفاض في القيمة هو أن حجم البيانات المرسله كان منخفضاً ويساوي 10.5 KBytes، ومن ثم بدأت بالارتفاع التدريجي بسبب زيادة حجم البيانات المرسله، ومنه نستنتج أن تطبيق القاعدة

(Rule_1) قد حقق فعالية في استخدام عرض لحزمة.

بينما كانت في السيناريو الأول قد استهلكت كامل عرض الحزمة وهو 1 Mbps.

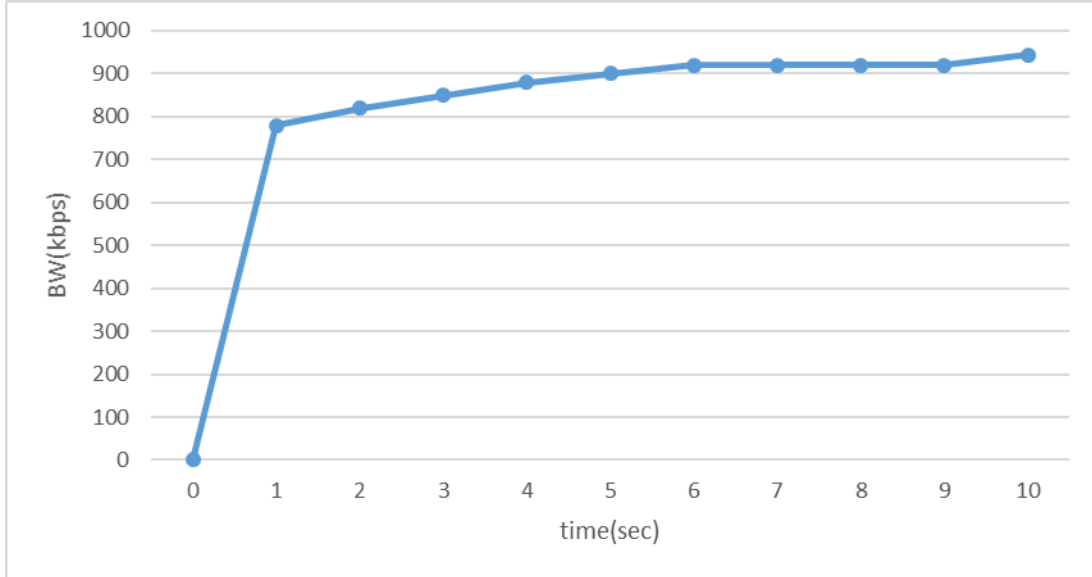
4-9 السيناريو الرابع:

نتحقق في هذه السيناريو من الآلية المقترحة حيث نستخدم الطوبولوجيا الموضحة في الشكل (4)، وننفذ التطبيق iperf وفق المحددات الموضحة في الجدول (5).

الجدول (5): محددات السيناريو الرابع

Scenario	Server	Client	Protocol	Port	Bandwidth	Interval
4 th . Scenario	H1 (IP:10.0.0.1)	H2 (IP:10.0.0.2)	UDP	5202	1 Mbps	1 sec

يوضح الشكل (10) نتائج تنفيذ السيناريو الرابع.



الشكل (10): عرض الحزمة المستخدم في السيناريو الرابع

نجد من الشكل (10) أن جميع الرزم التي تطابق القاعدة الثانية (Rule_2)، لم تتجاوز عرض الحزمة فيها القيمة 1000 kbps ولم تكن أقل من القيمة 700 kbps، حيث ازداد استهلاك عرض الحزمة تدريجياً خلال زمن التجربة لأن كمية البيانات المرسله في بداية التجربة تساوي وسطياً 112 Kbytes ومن ثم ازدادت بالتدرج. بينما كانت في السيناريو الثاني قد استهلكت كامل عرض الحزمة وهو 1 Mbps، مما يؤكد فعالية الآلية المقترحة في استهلاك عرض الحزمة.

نجد من نتائج تنفيذ هذه الآلية باستخدام المتحكم RYU قد ساهم بتوفير عرض الحزمة، حيث تم الحد من استخدامها (bandwidth limiting) من خلال توجيه كل تدفق في المسار المناسب وفقاً للقواعد التي تم إعدادها من قبل المتحكم. يتم إنجاز هذه الطريقة وتثبيت عرض الحزمة ضمن القيم المعرفة في الأرتال بشكل ستاتيكي، فهي طريقة معتمدة على التدفق (per-flow)، وتتطلب حجز عرض الحزمة بشكل مسبق.

إن هذه الطريقة تسبب انخفاض التأخير الزمني لوصول الرزم نحو هدفها، لكونها تخصص عرض الحزمة المناسب لكل رزمة من البيانات وفقاً لحجمها، حيث أن اختلاف حجم الرزم المرسله يسبب اختلاف في استهلاك عرض الحزمة.

الاستنتاجات والتوصيات:

- قدمنا في هذا البحث طريقة تمكّن المتحكم RYU من تحقيق جودة الخدمة من خلال الاستخدام الأفضل لعرض الحزمة مما يحسن من أداء الشبكة بشكل عام، وذلك من خلال تقديم طريقة تحقق مفهوم الأرتال وإسناد عرض الحزمة المناسب لكل رتل وفق عدة متطلبات مثل نوع البروتوكول المستخدم والمنفذ الذي يتم من خلاله تبادل الرزم.
- يساعد هذا البحث في تحسين استخدام عرض الحزمة في شبكات SDN، وذلك لتأمين جودة خدمة للتطبيقات التي تتطلب عرض حزمة كبير مثل تطبيقات الوسائط المتعددة.
- ومن خلال هذا البحث يمكننا تنفيذ هذا المقترح بواسطة متحكم آخر مثل Floodlight، إضافة إلى إجراء تحسينات أخرى لضمان جودة الخدمة بحيث تكون الخدمة ديناميكية وفقاً لمتطلبات التدفق الحالي دون الحاجة لإجراء إعدادات سابقة ضمن المتحكم وباقي الأجهزة.

References:

- [1] RANA D.; DHONDIYAL S.; CHAMOLI S. *Software Defined Networking (SDN) Challenges, issues and Solution*. International Journal of Computer Sciences and Engineering (IJCSSE), Vol.7, Issue-1, Jan., 2019.
- [2] RAJU V. *SDN CONTROLLERS COMPARISON*. Science Globe International Conference, India, 10th June, 2018.
- [3] JADHAV B.; SAQUIB Z.; PAWAR S. *ISSUES AND PARAMETERS FOR IMPROVING QoS AND PERFORMANCE IN SDN*. International Journal of Advances in Electronics and Computer Science, Vol.4, Issue 7, July, 2017.
- [4] KORYACHKO V.; PEREPELKIN D.; IVANCHIKOVA M.; BYSHOV V.; TSYGANOV I. *Analysis of QoS Metrics in Software Defined Networks*. 6th MEDITERRANEAN CONFERENCE ON EMBEDDED COMPUTING, 11-15 June, 2017.
- [5] DECUSATIS C.; CARRANZA A.; CACERES J. *Modeling Software Defined Networks using Mininet*. 2nd. International Conference on Computer and Information Science and Technology (CIST), 2016.
- [6] <https://iperf.fr/iperf-doc.php>
- [7] Ahmad A.; Mohammad A. A Study of OpenFlow Protocol and POX Controller in Software Defined Networks(SDN) Using Mininet. Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series. Vol(41) No(1) 2019, 535-553.
- [8] Sahoo K.; Mishra, m.; Sahoo S.; Sahoo B. *Software Defined Network: The Next Generation Internet Technology*. IJ. Wireless and Microwave Technologies, 2017.
- [9] LOPEZ P.; GEA J.; MARTINEZ F.; SANAHUJA J.; CRUZ A. *Host Discovery Solution: An Enhancement of Topology Discovery in OpenFlow based SDN Networks*. 3th International Joint Conference on e-Business and Telecommunications (ICETE), 2016.
- [10] MAMUSHIANE L.; LYSKO A.; DLAMINI S. *A Comparative Evaluation of the Performance of Popular SDN Controllers*. 10th Wireless Days Conference (WD), 2018.
- [11] ASADOLLAHI S.; GOSWAMI B.; SAMEER M. *Ryu Controller's Scalability Experiment on Software Defined Networks*. IEEE International Conference on Current Trends in Advanced Computing (ICCTAC), 2018.

[12] ODOM, W.; HOGG, S. *CCNA Routing and switching ICND2 200-105 Official Cert Guide*. Cisco Press, USA, 2017,1452.

[13] GHYAR, C.; SHAHADE M.; BAMB S.; MANKAR V. *Basics of Quality of Services (QoS)*. IJSRST, Vol.4, Issue 7, 2018.

[14] KARAKUS M.; DURRESI A. *Quality of Service (QoS) in Software Defined Networking (SDN): A Survey*. Journal of Network and Computer Applications (JNCA), 2016.