مجلة جامعة طرطوس للبحوث والدراسات العلمية_ سلسلة العلوم الهندسية المجلد (٤) العدد (٢) 2020
Tartous University Journal for Research and Scientific Studies- Engineering Sciences Series Vol. (4) No. (2) 2020

تحسين موازنة الحمل في مراكز البيانات اعتماداً على الشبكات المعرّفة بالبرمجيات

لبنى علي*
لمى محمود**

(تاريخ الإيداع ١٨ / ٦ / ٢٠١٩ . قُبل للنشر ٢٧ / ٢ / ٢٠٢٠)

الملخّص

يواجه مشغلو الشبكات من المؤسسات الكبيرة ومقدمو الخدمات السحابيّة تحديات كبيرة، مع استمرار نمو مراكز البيانات وتطبيقاتها. حديثاً، إن نموذج الشبكات المُعرَّفة بالبرمجيات SDN (Software Defined Networks) جرى اقتراحه كأكثر الحلول الواعدة لمستقبل الإنترنت لكونها قدّمت فصلاً بين مستوى التحكم في الشبكة ومستوى البيانات، وسمحت ببرمجة مستوى التحكم لينفذ تطبيقات متعددة وتحقيق ذكاء الشبكة، لكن بقي الاستخدام الأمثل لموارد الشبكة من أجل تحسين الأداء يمثل تحديًا خاصاً ومُلّحاً.

يقدّم هذا البحث خوارزمية ديناميكية لموازنة الحمل في شبكات مركز البيانات المستندة إلى SDN. تتمثّل مهمة الخوارزمية في جدولة تدفقات الشبكة القادمة وفق العدد الحالي للتدفقات على المسارات، ومن بين المسارات التي تحوي أقل عدد من التدفقات تختار الخوارزمية المسار الأقل تحميلاً.

تمَّ التنفيذ باستخدام المتحكم Ryu والبروتوكول OpenFlow ومحاكي الشبكة Mininet. أُجريت المقارنة مع خوارزمية التوجيه متعدد المسارات متساوية الكلفة ECMP المستندة إلى SDN، توضح نتائج التقييم تفوق الخوارزمية المقترحة من حيث معدل النقل الفعلي ونسب استخدامية الوصلات من أجل نماذج مختلفة من حركة البيانات.

الكلمات المفتاحية: مراكز البيانات، الشبكات المُعرَّفة بالبرمجيات، موازنة الحمل الديناميكية، المتحكم Ryu، بروتوكول التدفق المفتوح.

* أستاذ مساعد في قسم هندسة تكنولوجيا المعلومات – كليّة هندسة تكنولوجيا المعلومات والاتصالات – جامعة طرطوس – سوريا.
** طالبة ماجستير– قسم هندسة تكنولوجيا المعلومات – كليّة هندسة تكنولوجيا المعلومات والاتصالات – جامعة طرطوس – سوريا.

مجلة جامعة طرطوس للبحوث والدراسات العلمية_ سلسلة العلوم الهندسية المجلد (٤) العدد (٢) 2020
Tartous University Journal for Research and Scientific Studies- Engineering Sciences Series Vol. (4) No. (2) 2020

Enhancing Load Balancing in Software Defined Networks based Data Center Networks

Loubna Ali*
Lama Mahmoud**

(Received 18 / 6 / 2019 . Accepted 27 / 2 / 2020)

ABSTRACT

Network operators from large enterprises and cloud service providers face significant challenges as data centers and its applications continue to grow, Software Defined Networks (SDN) has been proposed as the most promising solution for the future of the Internet by providing a separation between control and data planes of networks. Thus allows implementation of multiple applications and achieve network intelligence, but the optimal use of network resources to improve performance still a special and urgent challenge.

This research presents a dynamic load balancing algorithm in SDN-based data center networks. Our algorithm is designed to schedule incoming network flows according to the current number of flows on the paths. Among the paths with the lowest number of flows, the lowest path is chosen.

It was implemented using the Ryu controller, Openflow protocol, and Mininet emulator. Compared with the static routing algorithm and the SDN-based Equal cost multipath (ECMP) algorithm, the evaluation results show that the proposed algorithm has better throughput and enhanced link utilization ratio.

Keywords: Data Center Networks, Software Defined Networks, Dynamic load balancing, Ryu controller, Openflow protocol.

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

مقدمة:

تشكّل الاتجاهات الضخمة الناشئة في تقنيات المعلومات والاتصالات (مثل البيانات الاجتماعية والسحابة والبيانات الضخمة) تحديات جديدة أمام الإنترنت في المستقبل، وشهدت شبكات مراكز البيانات DCN تطوراً غير مسبوق على مدى السنوات القليلة الماضية.[1] [2]

تعرّف شبكة مراكز البيانات بأنّها شبكة تربط الخوادم من خلال وصلات عالية السرعة وأجهزة التبديل داخل مركز البيانات، وهي جسر للحوسبة الموزعة على نطاق واسع. في ظل الزيادة الضخمة وتغيير المتطلبات في حركة البيانات في الإنترنت ، يجب الأخذ بالحسبان مراقبة حركة البيانات وإدارتها نظراً لأن أي تعطل في الخدمة أو تقديم معلّمات جودة خدمة منخفضة سيؤدي إلى خسارة هائلة في الإيرادات[3] ومن هذا المنطلق يعد تحسين أداء شبكات مركز البيانات أمراً بالغ الأهمية.

إن الأساليب التقليدية القائمة على الإعداد اليدوي للأجهزة معقدة وعرضة للخطأ، ولا يمكنها الاستفادة الكاملة من قدرات الشبكة، جرى تقديم الشبكات المعرفة بالبرمجيات SDN مؤخراً باعتبارها واحدة من أكثر الحلول الواعدة لمستقبل الإنترنت، فقد قدّمت العديد من الميزات، وأتاحت استيعاب الابتكار، وأسهمت في حل العديد من المشاكل التي عانت منها الشبكات التقليدية؛ مثل اختلاف أنظمة الإدارة بين المصنعين، وعملية الإعداد اليدوي للأجهزة، واتخاذ قرارات التوجيه محليّاً.[1] [4]

هدف البحث وأهميته:

تكمن أهمية البحث في كونه يسعى إلى تقديم حلول متقدمة في موازنة الحمل وجدولة التدفقات على المسارات المتاحة في مراكز البيانات وبالتالي تحسين استخدام الموارد وتحسين الأداء.

يقترح هذا البحث خوارزمية ديناميكية لموازنة الحمل في شبكات مركز البيانات المستندة إلى SDN، تتمثل مهمة هذه الخوارزمية في جدولة تدفقات الشبكة القادمة وفق العدد الحالي للتدفقات على المسارات؛ ومن بين المسارات التي تحوي أقل عدد من التدفقات تختار الخوارزمية المسار الأقل حملاً. وتتضمن ما يأتي:

- الاعتماد على نظرة شاملة لحالة الشبكة.
- كشف التدفقات الكبيرة وإيجاد أفضل المسارات لها؛ كونها السبب الرئيس في الازدحام.

طرق البحث ومواده:

يبدأ هذا البحث بإجراء دراسة مرجعية عن تقنيات موازنة الحمل المستخدمة في الشبكات المعرفة بالبرمجيات وكيفية تطبيقها في مراكز البيانات.

اعتمدت الدراسة العملية في البحث على المحاكي Mininet الذي يسمح بإنشاء شبكة كاملة على جهاز واحد، كما يشغل تعليمات برمجية حقيقية على أجهزة الشبكة الافتراضية، ويعدّ المحاكي الأكثر كفاءة ودقة في مجال SDN. كما استخدمت أداة iperf (Internet Performance Measurement tool) التي تستخدم لتوليد تدفقات البيانات وقياس الأداء.

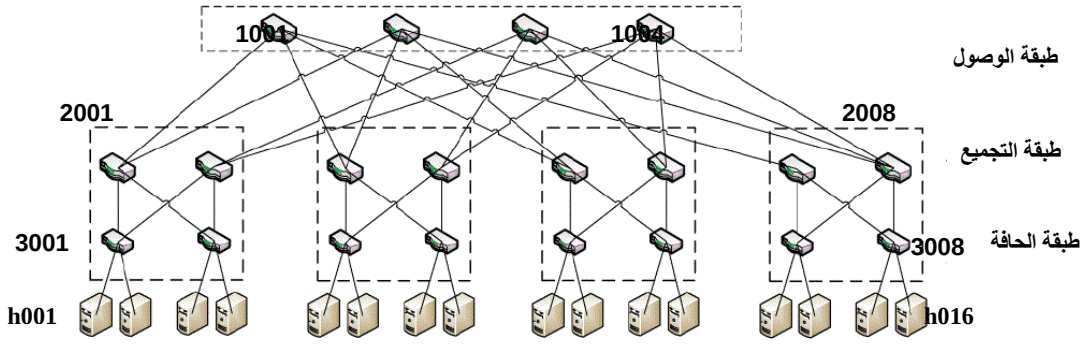
1 –شبكة مركز البيانات:

تُشكَّل مراكز البيانات اليوم من عشرات الآلاف من الأجهزة ، ونظراً لنوعية الخدمات التي تقدمها هذه المراكز فإنها حساسة جداً من ناحية الأداء، كما أنّ حركة البيانات فيها تتميز بوجود أحمال متغيرة باستمرار [5]. وتحوي بعض

التدفقات على كمية كبيرة من البيانات، وتتطلب عرض حزمة عالياً لإرسالها وتدوم لفترة طويلة. تجعل هذه الخصائص بنية الشبكة وخوارزميات التوجيه التقليدية فيها غير قابلة للتطبيق. [3]

اعتمدنا في هذا البحث طبولوجيا Fat Tree المبينة في الشكل (1) والتي تحظى بشعبية كبيرة لبناء شبكات مراكز البيانات، وتعدّ الاختيار الأفضل لتحقيق وثوقية عالية [6][7].

تحتوي هذه الطبولوجيا مسارات متعددة بين المضيفين، يمكنها توفير نطاق ترددي أكبر من الشجرة ذات المسار الواحد التي لها عدد العقد نفسها، وتتيح ميزة تعدد المسارات لهذه الشبكات فرصاً لتوزيع حركة مرور البيانات على مكونات الشبكة المختلفة. وهي شجرة ثلاثية الهرمية تتكون من مبدلات على طبقات التجميع وطبقة الوصول، يتصل المضيفون بالمبدلات الموجودة على طبقة الحافة Edge. [2]



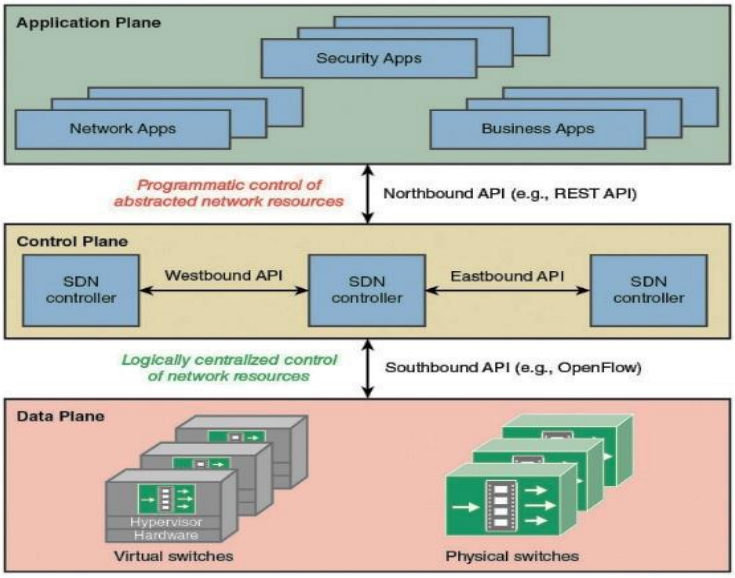
الشكل(1): الطبولوجيا الشجرية Fat Tree

برغم أن طبولوجيا Fat Tree توفر اتصالاً غنياً، فإن وجودها بمفرده لا يضمن أداء شبكة عالياً؛ حيث تلعب آلية التوجيه دوراً حاسماً في الأداء.

2- الشبكات المعرفة بالبرمجيات:

تعتمد الشبكات المعرفة بالبرمجيات على فصل مستوى التحكم عن مستوى توجيه البيانات، وتوفير قابلية البرمجة لتطوير تطبيقات الشبكة، لجعل التحكم بالشبكة مركزي منطقياً. [3]

يبين الشكل (2) البنية المرجعية لـSDN [8] كما عرّفتها مؤسسة الشبكات المفتوحة (Open Networks ONF Foundation) وهي منظمة غير ربحية مختصة بوضع معايير SDN تطويرها، يقودها مجلس إدارة من سبع شركات عالمية.



شكل(2): البنية المرجعية لنموذج الشبكات المعرّفة بالبرمجيات SDN

يتكون هذا النموذج من الطبقات الآتية:

1. طبقة التطبيقات (Application Layer): تقوم بفرض منطق التحكم الذي يترجم إلى قواعد تحكم عمل طبقة البيانات، ومن أمثلة التطبيقات: بروتوكولات التوجيه، قوائم التحكّم بالوصول، مراقبة الشبكة.

2. طبقة التحكم (Control Layer): تمثل المتحكم controller وهو أساس تقنية SDN، يسمى نظام تشغيل الشبكة، يؤمن خدمات أساسية مثل تحصيل معلومات الطبولوجيا والتعرّف إلى الأجهزة المتصلة.

3. طبقة البنية التحتية (طبقة المعطيات Data Layer)؛ تتألف من مجموعة من التجهيزات الشبكية التي تقوم بدور التمرير من دون القدرة على اتخاذ قرارات التحكم. تدعم واجهات مفتوحة وموحدة مثل OpenFlow مما يتيح لطبقة التحكم إدارة أجهزة متمايزة المصنعين وهذه إحدى الأمور التي تنقص الشبكات التقليدية.[9] [10]

3 –بروتوكول التدفق المفتوح (OpenFlow Protocol):

يعدّ البروتوكول OpenFlow أول واجهة قياسية والأكثر شيوعاً واستخداماً في متحكمات الشبكات المعرفة بالبرمجيات للتواصل والتحكم بالمبدلات، من خلاله يتعلم المبدل معلومات التوجيه من المتحكم ومن ثم يمرر رزم البيانات اعتماداً على هذه المعلومات.[10] [11]

4– موازنة الحمل (Load balancing):

موازنة الحمل هي واحدة من أكبر المشاكل المتعلقة بالشبكات الحاسوبية. يمكن أن يكون الحمل على شكل حمل الذاكرة أو سعة وحدة المعالجة المركزية أو تحميل الشبكة أو حتى التأخير. من أجل تحسين أداء النظام إلى جانب استخدام الموارد، من المفترض أن يشارك موازن الحمل باستمرار عبء العمل على جميع العقد في النظام الموزع.[8] يمكن تنفيذ موازنة الحمل برمجياً أو عتادياً.

الأهداف الرئيسة لموازنة الحمل:

–تحسين أداء النظام.

–الحفاظ على استقرار النظام.

-بناء نظام متسامح مع الأخطاء.

-القدرة على إجراء تعديلات في المستقبل.

4-1 موازنة حمل الوصلة Link Load balancing

عندما يتم تعيين تدفقات متعددة إلى نفس الوصلة، قد تعاني بعض تدفقات البيانات من مشاكل مثل الازدحام وتأخير الإرسال الطويل، يتم تحسين استخدام الوصلات باستخدام إستراتيجية جدولة معينة لتوزيع تدفقات البيانات على مسارات مختلفة. [12]

- تعد تقنية التوجيه متعدد المسارات متساوية الكلفة ECMP(Equal Cost Multipath) إحدى أكثر التقنيات استخداماً لموازنة حمل الوصلات، وهي تقنية توجيه سهلة وعملية للشبكات؛ وتهدف للحصول على عدة مسارات متساوية الكلفة إلى الوجهة نفسها، وتوجيه البيانات عبر هذه المسارات المتعددة، مما يؤدي إلى تخفيض الوقت المستغرق لإرسال البيانات. في حال وجود مسارات مرشحة متعددة، تستخدم آلية لتحديد المسار الذي سيتم استخدامه لتسليم البيانات. هناك عدة طرق لتحديد المسارات، من بينها: الاختيار العشوائي، الآلية الدورية، وModulo-N Hash. [13]

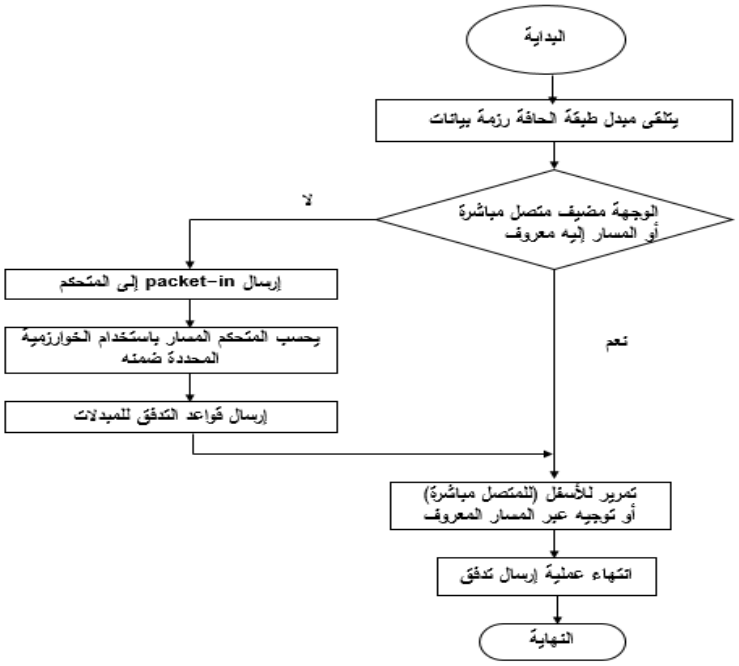
تستخدم ECMP آليات ثابتة لاختيار المسار ولا تأخذ بالحسبان حالة المسار أو متطلبات التدفق. [14]

- اقتُرحت بعض الخوارزميات اختيار أفضل مسار من بين المسارات البديلة بطريقة ديناميكية وشاملة، اقترح الباحثون في [15] خوارزمية يتم فيها اختيار المسار بواسطة منصة التحكم المركزية اعتماداً على آلية التقييم الضبابية (FSEM: Fuzzy Synthetic Evaluation Mechanism)، والتي تأخذ بالحسبان عدة بارامترات. كما اقترح الباحثون في [3] آلية لموازنة الحمل تعتمد على الخوارزمية الجينية في انتقاء المسار، وجرى قياس أدائها من خلال قدرتها على موازنة الحمل فقط، كما لم يُطبَّق أياً منهما على بنية ضخمة متل شبكة مركز البيانات حيث تتطلب موارد معالجة كبيرة وتأخيرات. نظراً لأن الوقت الذي يتم فيه حساب المسار الأمثل يضاف إلى تأخير التدفق، وقد يمنح المسار الأمثل مكسبًا في أداء الإرسال أقل من تأخير الانتظار الطويل.

تناول هذا البحث إجراء مقارنة مع خوارزمية ECMP كونها الخوارزمية المعيارية والأكثر استخداماً وتطبيقاً في مجال موازنة الحمل في الشبكات التقليدية وأيضاً في SDN.

5- الخوارزمية المقترحة:

يبين الشكل (3) الآلية العامة للتوجيه في الشبكات المعرفة بالبرمجيات؛ حيث يجري اختيار مسار إرسال البيانات وفق الخوارزمية الموجودة في المتحكم، إلا إذا كان المضيف متصلاً مع المصدر مباشرة على المبدل نفسه، أو أن المبدل قد تعلّم فعلياً مسار الإرسال وخزّنه في جداول التدفق الخاصة به.



الشكل(3): آلية التوجيه في SDN

تأخذ خوارزمية موازنة الحمل المقترحة بالحسبان محاولة الجمع بين التنفيذ غير المعقد والأداء الجيد، كما تراعي خصائص حركة البيانات في مراكز البيانات؛ حيث تحوي بعض التدفقات كميةً كبيرة من البيانات، وتتطلب عرض حزمة عالياً لإرسالها، وتدوم مدَّةً طويلة.

يستخدم المتحكم ryu الخوارزمية المقترحة المكتوبة بلغة بايثون في إيجاد المسارات وجدولة التدفقات خلالها وتتضمن المراحل الآتية:

- الحصول على كافة المسارات الأقصر الممكنة للوصول إلى الوجهة K-shortest path.
- كشف التدفقات الكبيرة على المسارات المرشحة من المرحلة السابقة، واختيار المسارات ذات العدد الأقل من التدفقات الكبيرة (يتم حساب عدد التدفقات الكبيرة على مسار معيّن بالاعتماد على العدد الأعظمي للتدفقات الكبيرة على طول وصلات هذا المسار).
- الحصول على معلومات تحميل المسارات ذات العدد الأقل من التدفقات الكبيرة، واختيار المسار الذي يمتلك أعلى سعة متاحة (يتم حساب معلومات سعة مسار معيّن بالاعتماد على السعة المتاحة الأصغرية على طول وصلات هذا المسار).
- المسار الذي يتمّ إيجاده هو عبارة عن قائمة من المبدلات من المصدر إلى وجهة الرزمة. يم تحويل معلومات المسار إلى قواعد تدفق وإرسالها، حيث إنّ كلّ مبدّل موجود ضمن المسار المختار يجب أن تكون لديه مُدْخَلات التدفق الضرورية لإقامة اتصال بين كل نقطتين.
- إذاً تعتمد الخوارزمية المقترحة على كشف التدفقات الكبيرة لتوجيهها إلى مسار غير مزدحم لمنع تصادم تدفقين كبيرين على المسار نفسه، كما تراعي معلومات الحمل المسار.

5-1 مراقبة احصائيات المنافذ:

تتطلب الخوارزمية المقترحة أحدث سعة متاحة للمسار وهذا لا يمكن استرجاعه بشكل مباشر من المتحكم ولا من المبدلات. يستعلم مكوّن المراقبة كلّ المبدلات حول إحصائيات المنافذ، ويحسب السعة الحرّة من خلال عدد البايتات المرسلة على المنفذ خلال فاصل زمني معين.

بداية كلّ دورة مراقبة يرسل المتحكم رسالة OFPPortStatsRequest لكلّ المبدلات، طالباً رسالة OFPPortStatsReply التي تحوي معلومات احصائية عن المنافذ. يحتفظ المتحكم بمجموعة المنافذ وعدد وحدات البايت التي أرسلها كلّ منفذ في دورة المراقبة الأخيرة ضمن بنيّة بيانات من نوع قاموس (dictionary of dictionary) من الشكل: S = { switchID:{ portnumber:free_bw,},}

بمجرد رد المبدل على المتحكم برسالة الإجابة، سيقوم المتحكم بحساب السعة الحرّة free bandwidth وتحديث قيمتها لجميع المنافذ في S وفقاً للمبدّل والمنفذ.

تُحسب السعة الحرة free_bw لكلّ منفذ بالاعتماد على حمل المنفذ والسعة الإجمالية باستخدام العلاقة (1).

$$Free\_bw=\max(capacity - load\ ,\ 0) \qquad (1)$$

يقاس حمل المنفذ (مُقدراً بالبايت) باستخدام العلاقة (2).

$$Load=\frac{tx\_bytes\_now-tx\_bytes\_pre}{p} \qquad (2)$$

حيث:

- Tx_bytes_now عدد وحدات البايت المرسلة حتى اللحظة.
- Tx_bytes_pre عدد وحدات البايت المرسلة في آخر دورة مراقبة.
- p المدة الزمنية بين القياسين.

5-2 كشف التدفقات الكبيرة:

تعدّ من القضايا الرئيسة التي يتعيّن حلّها في الخوارزمية المقترحة، يتحقق اكتشاف التدفقات الكبيرة في الوصلات من خلال أخذ إحصائيات عن كل مُدْخَل تدفق لكل مبدّل باستخدام رسالة الطلب OFPFlowStatsRequest. في النموذج الأولي للتنفيذ، يتم تمييز تدفق الشبكة المقابل لمُدْخَل التدفق الذي لا يقلّ متوسط معدّل إرساله عن 5% من سعة الوصلة على أنّه تدفق كبير.

5-3 الحصول على المسار الأنسب:

تتضمن عملية الحصول على المسار الأنسب الخطوات الآتية:

١- استنتاج عدد التدفقات على طول كل مسار من المسارات المتاحة: تحتاج الخوارزمية لحساب عدد التدفقات على طول مسار معين، ويكون ذلك وفق العلاقة (3) كما يأتي:

- مقارنة عدد التدفقات على كلّ منفذ بدءً من أول وصلة في المسار (وفق آلية كشف التدفقات الكبيرة).
- اعتبار القيمة الأعلى الناتجة هي عدد التدفقات الكبيرة.

$$fnum\ (path) = \max fnum_i\ (links\ along\ path) \qquad (3)$$

- استنتاج المسار الأقل تحميلا: تحتاج الخوارزمية لحساب السعة المتاحة للمسار وذلك بمقارنة السعة المتاحة على المنافذ على طول المسار؛ وابتداءً من أول وصلة في المسار (وفق العلاقة (4)). يتم اعتبار القيمة الأدنى هي

السعة المتاحة ومتابعة المقارنة مع أدنى قيمة تم الحصول عليها، بآلية مشابهة للآلية السابقة حتى الوصول إلى أدنى سعة متاحة على طول المسار واعتبارها السعة المتاحة للمسار بأكمله.

$$free\_bw\ (path) = \min free\_bw_i\ (links\ along\ path) \qquad (4)$$

النتائج والمناقشة:

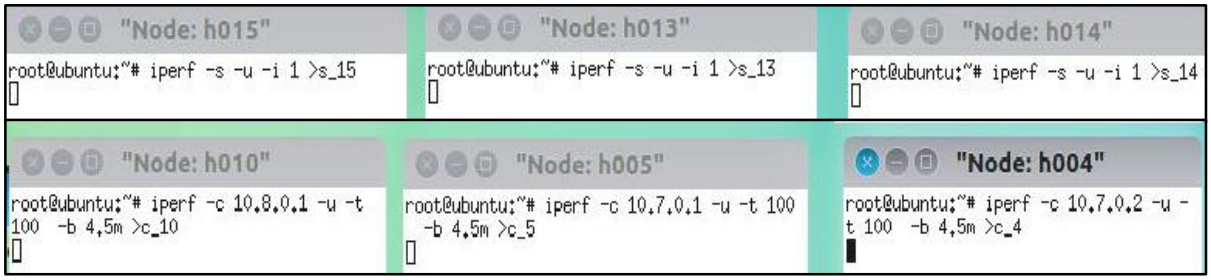
قمنا بتشغيل الطبولوجيا المخصصة في Mininet، وتتكون من 20 مبدلاً OpenFlow، 16 مضيفاً متصلين بمبدلات الحافة Edge، وصلات الشبكة بسعة 8 Mbit، متحكم SDN Remote controller على المنفذ: 6633.

سيناريوهات العمل:

4-1 السيناريو الأول: يهدف إلى حساب خسارة البيانات عند وجود ثلاثة تدفقات كبيرة في الشبكة:

سنقوم بتوليد التدفقات ثلاثة تدفقات في الشبكة من نوع udp (4.5 mbps) ومدة كل تدفق 100 ثانية كالآتي:

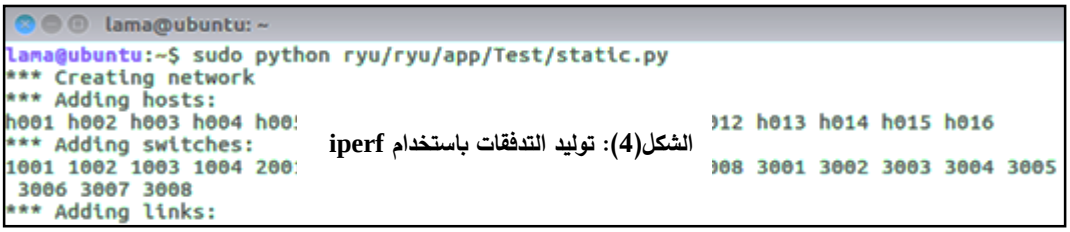
- تشغيل العقدة h014 (10.7.0.2) كـ iperf server والعقدة h004 (10.2.0.2) كـ iperf client
- تشغيل العقدة h013 (10.7.0.1) كـ iperf server والعقدة h005 (10.2.0.1) كـ iperf client



- تشغيل العقدة h015 (10.8.0.1) كـ iperf server والعقدة h010 (10.5.0.2) كـ iperf client

أولاً من أجل خوارزمية التوجيه عن طريق مسار واحد:

الطبولوجيا وخوارزمية التوجيه مبرمجين ضمن ملف static.py وعند تشغيله سيتم بناء طبولوجيا Fat Tree تستخدم تقنية التوجيه المعتمدة على عنوان الوجهة فقط، والتي تستخدم مساراً واحداً لكل زوج من العقد.



الشكل(4): توليد التدفقات باستخدام iperf

الشكل(5): تشغيل الطبولوجيا مع حالة مسار واحد

قمنا بتوليد التدفقات سابقة الذكر وراقبنا مسارات إرسالها باستخدام برنامج wireshark فكانت ممثلة بالمبدل(المنفذ):

مسار التدفق الأول:

3002(eth4) - 3002(eth1) - 2001(eth4) -2001(eth1) **-1001(eth1)** – **1001(eth4) – 2007(eth1) - 2007(eth3) - 3007(eth1)** - **3007(eth4)**

مسار التدفق الثاني:

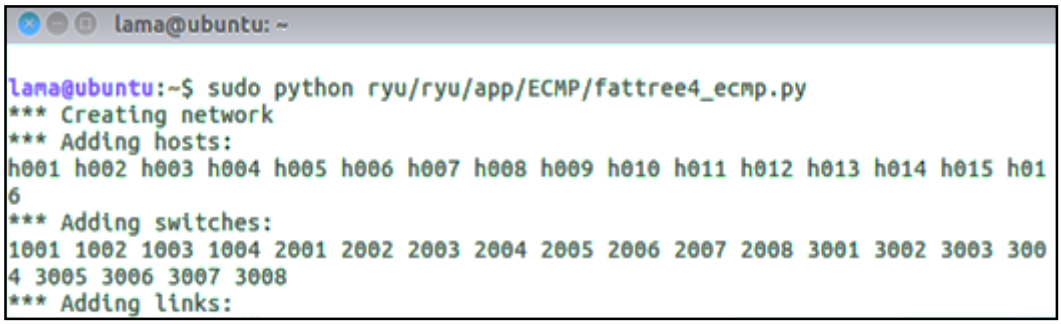
3003(eth3) - 3003(eth1) - 2003(eth3) -2003(eth1) **-1001(eth2)** – **1001(eth4) – 2007(eth1)** - **2007(eth3) - 3007(eth1)** - **3007(eth3)**

مسار التدفق الثالث:

3005(eth4) - 3005(eth1) - 2005(eth3) -2005(eth1) **-1001(eth3)** – **1001(eth4) – 2007(eth1)** - 2007(eth4) - 3008(eth1) - 3008(eth3)

ثانياً من أجل خوارزمية ECMP:

الطبولوجيا وخوارزمية ECMP مبرمجين ضمن ملف fattree4_ecmp.py وعند تشغيله سيتم بناء طبولوجيا Fat Tree تستخدم تقنية التوجيه ECMP، وجرى تنفيذ جميع الأساليب في هذه التجربة باستخدام إدخالات التدفق وجدول المجموعة Group table من نوع select الذين يتيحهم البروتوكول OpenFlow v1.3. لتأمين تحقيق سلوك مشابه تماماً لطريقة اختيار المسار التي تتم في ECMP.



الشكل(6): تشغيل الطبولوجيا مع خوارزمية ECMP

كانت المسارات كالآتي:

مسار التدفق الأول:

3002(eth4) - 3002(eth2) - 2002(eth4) -2002(eth1) **-1003(eth1)** – **1003(eth4) – 2008(eth1)** - **2008(eth3)** - 3007(eth2) - 3007(eth4)

مسار التدفق الثاني:

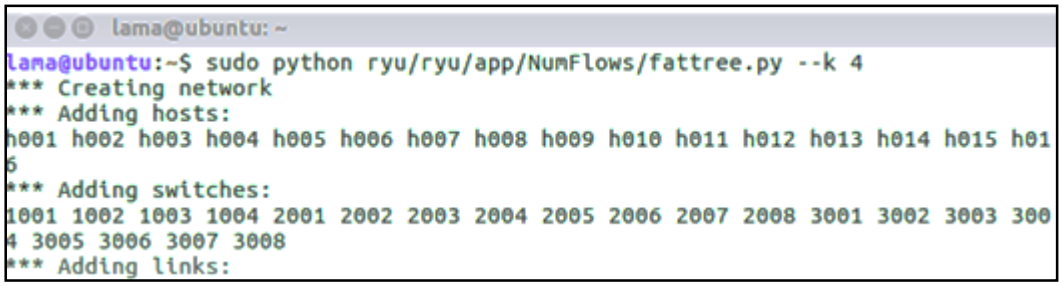
3003(eth3) - 3003(eth2) -2004(eth3) -2004(eth1) **-1003(eth2)** – **1003(eth4) – 2008(eth1)** - **2008(eth3)** - 3007(eth2) - 3007(eth4)

مسار التدفق الثالث:

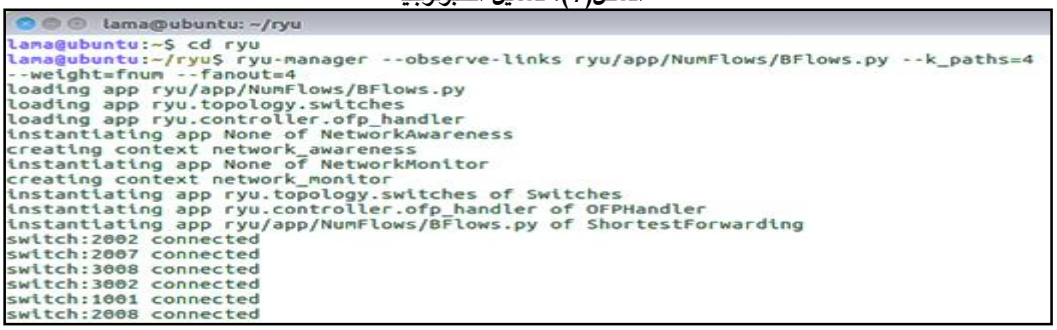
3005(eth4) - 3005(eth2) - 2006(eth3) -2006(eth1) **-1003(eth3)** –**1003(eth4) – 2008(eth1)** - 2008(eth4) - 3008(eth2) - 3008(eth3)

ثالثاً من أجل الخوارزمية المقترحة:

ملف الطبولوجيا fattree.py يتم تشغيله ضمن Mininet يحوي الطبولوجيا المستخدمة في الحالتين السابقتين نفسها؛ لكن في هذه الحالة سيجري تشغيل الخوارزمية NumF من خلال ملف منفصل يعمل كتطبيق على متحكم Ryu Remote controller موجود على الجهاز نفسه ومتصل عبر المنفذ: 6633



الشكل(7): تشغيل الطبولوجيا



الشكل(8): تشغيل المتحكم مع الخوارزمية المقترحة

كانت المسارات كالآتي:

مسار التدفق الأول:

3002(eth4) - 3002(eth1) - 2001(eth4) -2001(eth1) -1001(eth1) – 1001(eth4) – 2007(eth1) - 2007(eth3) - 3007(eth1) - 3007(eth4)

مسار التدفق الثاني:

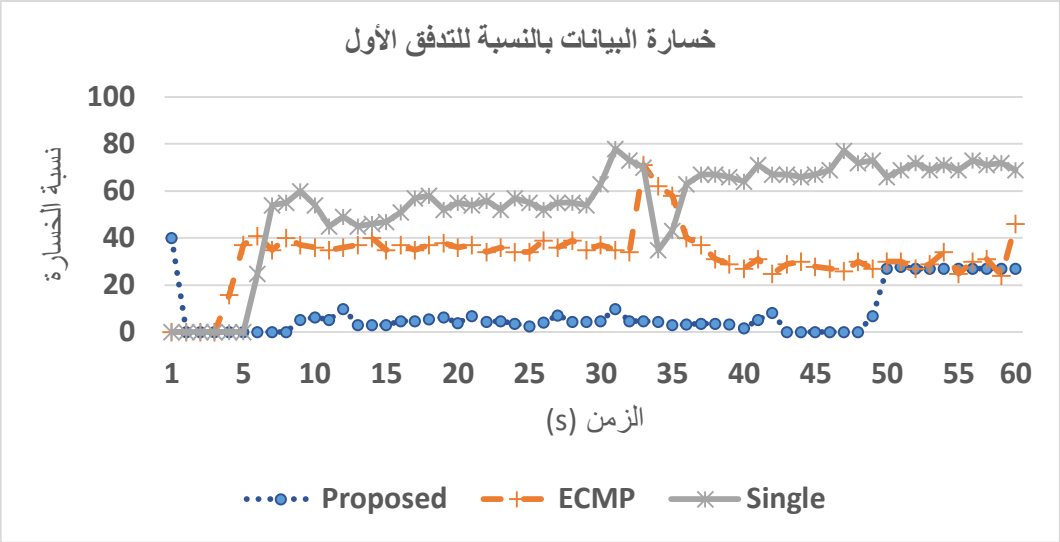
3003(eth3) - 3003(eth2) - 2004(eth3) -2004(eth1) -1003(eth2) – 1003(eth4) – 2008(eth1) - 2008(eth3) - 3007(eth2) - 3007(eth3)

مسار التدفق الثالث:

3005(eth4) - 3005(eth1) - 2005(eth3) -2005(eth2) -1002(eth3) – 1002(eth4) – 2007(eth2) - 2007(eth4) - 3008(eth1) - 3008(eth3)

1-1 خسارة البيانات:

تعطي الأداة iperf تقريراً يتضمن خسارة البيانات اللحظية لكل تدفق، بالنسبة إلى التدفق الأول في الشبكة H4-H14 كان كما في الشكل (9):

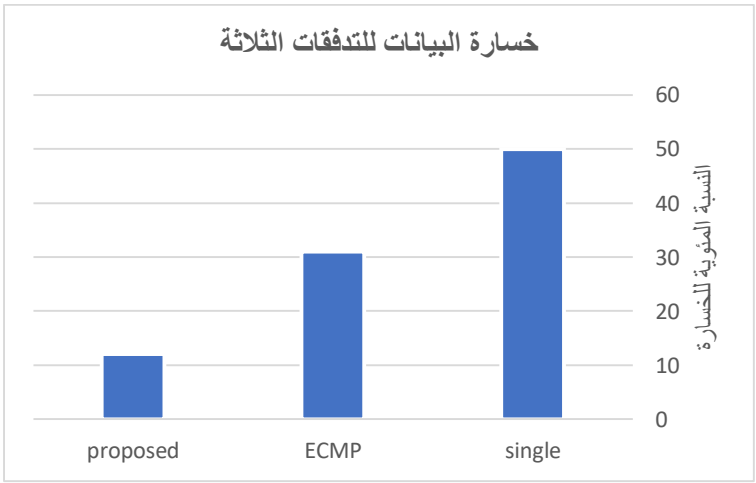


الشكل(9): خسارة البيانات بالنسبة للتدفق الأول

نلاحظ أنه في اللحظة الأولى، و يانات في حالة الخوارزمية المقترحة، سببه قيام المتحكم بالحسابات التي يطبقها اختيار المسار، بينما لا تتم هذه الحسابات في حالة single و ECMP حيث توجد قواعد استباقية تحكم العمل.

مع مرور الوقت ودخول التدفقات الأخرى يتم جدولتها إلى مسارات غير مزدحمة في حالة الخوارزمية المقترحة لذا كان فقد البيانات الأقل. بينما في حالة single و ECMP. يزداد الازدحام ممّا يؤدي لزيادة فقد البيانات في الشبكة بسبب تداخل التدفقات على بعض المسارات، كما اعطى استخدام مسار واحد فقد البيانات الأعلى كونه سبب ازدحاماً في الوصلات أكثر من حالة ECMP.

يلخص الشكل(10) متوسط خسارة البيانات للتدفقات الثلاثة:



الشكل(10): متوسط خسارة البيانات للتدفقات الثلاثة

4-2 السيناريو الثاني: يهدف إلى حساب المعدل الفعلي لنقل البيانات عبر الشبكة، واستخدامية الوصلات، وذلك من أجل نماذج مختلفة من حركة البيانات.

يحاكي هذا السيناريو وجود 16 تدفقاً بين مختلف المضيفين، و5 نماذج لحركة البيانات: عشوائية، 0.2_0.1، 0.3_0.2، 0.3_0.5، 0.1_0.8.

النموذج العشوائي: يتم اختيار المضيف الذي يبدأ التدفق والمضيف النظير بشكل عشوائي تماماً.

باقي النماذج: يتم اختيار مخدم iperf بشكل عشوائي واختيار النظير client وفق احتمالية نموذج الحركة مثلاً:

0.1_0.2: 10% من حركة البيانات تتم بين المضيفين الموجودين تحت مبدل طبقة Edge نفسه، 20% بين ... pod. كما هو موضح في الجدول (1).

الجدول(1): أزواج iperf المولّدة في النموذج 0.1_0.2

ضمن مبدل طبقة Edge	(h006 – h005)
ضمن الـ pod	(h015 – h014) , (h016 – h013) , (h008 –h005)
ضمن pods مختلفة	(h002 – h007) , (h010 – h008) , (h004 –h016) ,(h007 – h015) , (h011 – h016) , (h003– h013) , (h001 – h006) , (h013 – h007) , (h009 – h015) ,(h014 – h005) , (h012 – h006) , (h005 – h001)

تم استخدام إعدادات متطابقة تماماً للتدفقات في الحالات الثلاثة: استخدام مسار وحيد، ecmp، الخوارزمية المقترحة.

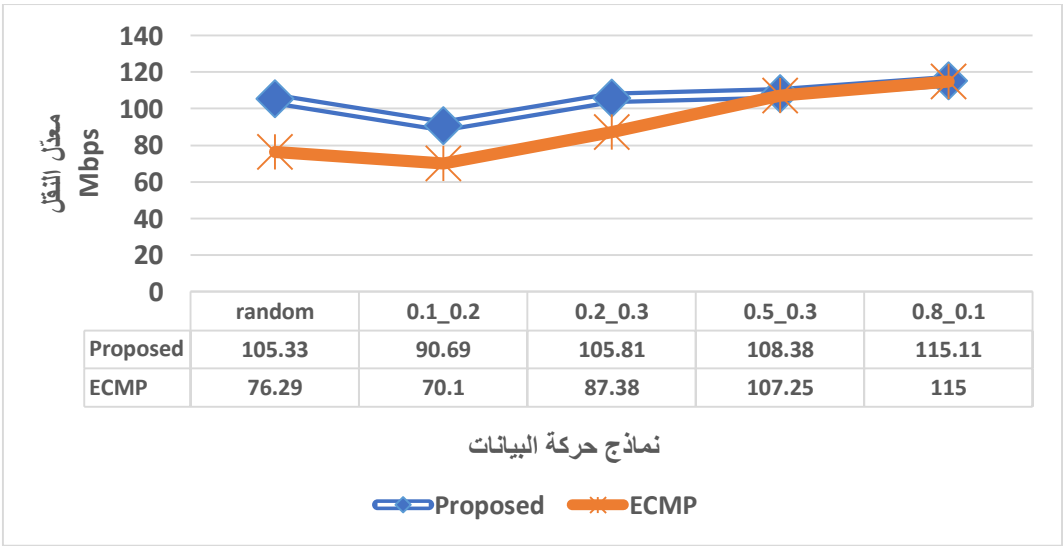
التدفقات من نوع tcp التي يولدها iperf افتراضياً.

المراقبة تبدأ بعدد 10 ثواني من توليد التدفقات (لتكون مستقرة) وتستمر المراقبة مدة 60 ثانية.

2-1 المعدل الفعلي لنقل البيانات:

جرى في هذا البحث قياس المعدل الفعلي لنقل البيانات الذي يعبّر عن كمية البتات المنتقلة بشكل صحيح عبر الشبكة خلال واحدة الزمن، والذي يقدّر بواحدة بت/ثا.

يبين الشكل (12) متوسط معدّل النقل الذي يعكس قدرة الشبكة على نقل حركة البيانات، وهو مؤشر الأداء الرئيس لهذه التجربة.

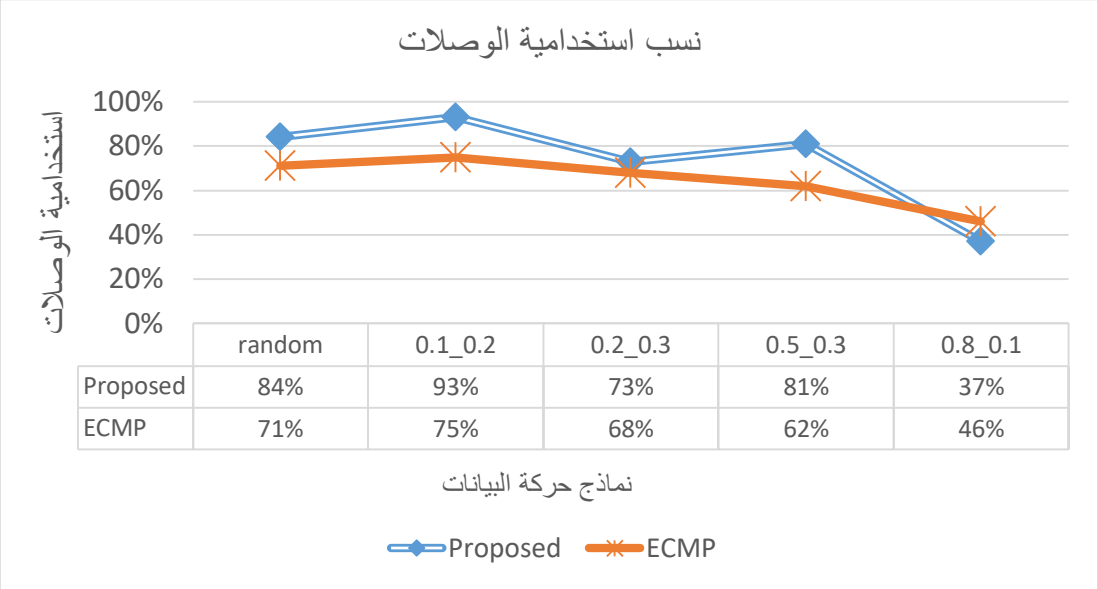


الشكل(12): متوسط معدل النقل من أجل نماذج مختلفة من حركة البيانات

بشكل عام، عندما تكون حركة البيانات تحت نفس مبدل طبقة Edge (أيضاً ضمن pod) منخفضة نسبياً، يكون فرق معدّل النقل الفعلي الناتج عن ECMP والخوارزمية المقترحة واضحاً بسبب ازدياد نسبة تصادم التدفقات الممكن حدوثه عند استخدام ECMP، وعندما تكون حركة البيانات ضمن الـ pod مرتفعة نسبياً، يكون معدل النقل متقارباً في الحالتين كون المسارات تصبح أكثر تحديداً.

4-2-2 استخدامية الوصلات:

يبين الشكل (13) استخدامية وصلات الشبكة حيث تشير استخدامية الوصلات إلى النسبة بين عدد الوصلات المستخدمة فعلياً وإجمالي الوصلات، وتعكس استخدام الموارد وتوازن الحمل في الشبكة.



الشكل (13): استخدامية الوصلات من أجل نماذج مختلفة من حركة البيانات

عندما تكون حركة البيانات تحت نفس مبدل طبقة الحافة منخفضة نسبيا، يكون عدد الوصلات المستخدمة في الشبكة كبيراً، وعندما تكون مرتفعة نسبياً، يكون عدد الوصلات المستخدمة صغيراً.

نلاحظ من الشكل (13) أن الخوارزمية المقترحة حققت استخدامية أعلى نسبياً، ويظهر الفرق بشكل أوضح عندما تكون النسبة الأكبر من حركة البيانات بين مضيفين من مبدلات مختلفة، حيث تزداد المسارات المتاحة وتزداد فرص الخوارزمية في إيجاد الأفضل بينها.

5- الاستنتاجات والتوصيات:

- جرى في هذا البحث اقتراح خوارزمية لموازنة الحمل في شبكات مراكز البيانات المعتمدة على الشبكات المُعرَّفة بالبرمجيات، كما تمَّ دراسة وتنفيذ خوارزمية ECMP الأكثر شهرة واستخدام في هذا المجال والتي تعتمد جدولة ستاتيكية للتدفقات على المسارات المتاحة ومقارنة النتائج في سيناريوهين مختلفين.
- جرت برمجة الخوارزميتين ECMP والمقترحة بلغة بايثون واستخدام OpenFlow v1.3 كبروتوكول تواصل بين المتحكم Ryu ومستوى البيانات ممثلاً بالمبدلات.
- بيَّنت النتائج أنَّ الخوارزمية المعتمدة على عدد التدفقات الكبيرة تفوقت بأدائها من حيث معدل النقل، وفقد البيانات على خوارزمية ECMP، كونها حسّنت بشكلٍ عامّ من استخدام الوصلات المتاحة في الشبكة.

- استُخدمت شبكة Fat Tree (K=4 pods)، ولكن لم يتم اختبار شبكة (K = 8)، لأنَّه لا يوجد سوى متحكم Ryu واحد في التجارب التي أجريت في هذا البحث، ولاختبار طبولوجيا Fat Tree بحجم (K = 8 pods). ويوصى باستخدام أكثر من متحكم في بنيّة موزّعة لمستوى التحكم (وممركزة منطقياً من خلال التعاون بين المتحكمات).

المراجع:

[1] Xia, W., Wen, Y., Foh, C. H., Niyato, D., & Xie, H. (2014). *A survey on software-defined networking. IEEE Communications Surveys & Tutorials*, 17(1), 27-51.

[2] Bilal, K., Malik, S. U. R., Khan, S. U., & Zomaya, A. Y. (2014). *Trends and challenges in cloud datacenters. IEEE cloud computing*, 1(1), 10-20.

[3] Chou, L. D., Yang, Y. T., Hong, Y. M., Hu, J. K., & Jean, B. (2014). *A genetic-based load balancing algorithm in openflow network. In Advanced Technologies, Embedded and Multimedia for Human-centric Computing* (pp. 411-417). Springer, Dordrecht.

[4]علي، لبنى. تحسين جودة أداء محددات إدارة الشبكات الموزعة باستخدام العميل البرمجي المتحرك. مجلة جامعة طرطوس للبحوث والدراسات العلمية. سلسلة العلوم الهندسية ،المجلد (١) ،العدد (١) ،٢٠١٧.

[5] Benson, T., Anand, A., Akella, A., & Zhang, M. (2009, August). *Understanding data center traffic characteristics. In Proceedings of the 1st ACM workshop on Research on enterprise networking* (pp. 65-72). ACM.

[6]Couto, R. S., Campista, M. E. M., & Costa, L. H. M. (2012, December). *A reliability analysis of datacenter topologies. In 2012 IEEE Global Communications Conference (GLOBECOM)* (pp. 1890-1895). IEEE.

[7]علي، لبنى. تصميم شبكة حكومية آمنة باستخدام تقنية MPLS/VPN. مجلة جامعة طرطوس للبحوث والدراسات العلمية. سلسلة العلوم الهندسية ،المجلد (٢) ،العدد (١) ،٢٠١٨.

[8] Ghosh,A.(2018),A study on load balancing techniques in SDN, International Journal of Engineering & Technology ,India, Vol 7, No 2.4

[9] محمد، أحمد. دراسة البروتوكول OpenFlow والمتحكم POX في الشبكات المعرفة بالبرمجة SDN باستخدام Mininet. مجلة جامعة تشرين للبحوث والدراسات العلمية. سلسلة العلوم الهندسية، المجلد (٤١) ،العدد (١) ،٢٠١٩.

[10] Kreutz, D., Ramos, F. M., Verissimo, P., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1), 14-76.

[11] Jammal, M., Singh, T., Shami, A., Asal, R., & Li, Y. (2014). *Software defined networking: State of the art and research challenges. Computer Networks*, 72, 74-98.

[12] Li, L., & Xu, Q. (2017, July*). Load balancing researches in SDN: A survey. In 2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC)* (pp. 403-408). IEEE.

[13] Rhamdani, F., Suwastika, N. A., & Nugroho, M. A. (2018, May). *Equal-Cost Multipath Routing in Data Center Network Based on Software Defined Network. In 2018 6th International Conference on Information and Communication Technology (ICoICT)* (pp. 222-226). IEEE.

[14] Zhang, H., Guo, X., Yan, J., Liu, B., & Shuai, Q. (2014, October). *SDN-based ECMP algorithm for data center networks. In 2014 IEEE Computers, Communications and IT Applications Conference* (pp. 13-18). IEEE.

[15] Li, J., Chang, X., Ren, Y., Zhang, Z., & Wang, G. (2014, September*). An effective path load balancing mechanism based on SDN. In 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications* (pp. 527-533). IEEE.