مجلة جامعة طرطوس للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (5) العدد(5) 2021

Tartous University Journal for Research and Scientific Studies - engineering Sciences Series Vol. (5) No. (5) 2021

تَقييمُ أداءِ خوارزميات موازنة الحمل في الشبكات المعرّفة بالبرمجيات (SDN)

أ. م. د محمد صبيح*

م. ايمان خليل**

(تاريخ الإيداع 4 /2 /2021 . قُبِل للنشر في 23 /6 /2021)

□ ملخّص □

نظراً للنمو السريع في استخدام الخدمات القائمة على الشبكة، يجب على الشبكات التعامل مع كمية هائلة من حركة البيانات، وخدمة الآلاف من المستخدمين. حيث أن وجود مخدم واحد غير قادر على التعامل مع مثل هذا الحمل الضخم. فكان الحل استخدام مخدمات متعددة مع موازن حمل يعمل بمثابة الواجهة الأمامية. يتلقى موازن الحمل الطلبات من مستخدمين متعددين واعتماداً على استراتيجيات موازنة الحمل المختلفة يقوم بتوزيع هذه الطلبات على عدد من المخدمات. تتنوع استراتيجيات موازنة الحمل بين المخدمات في الشبكات المعرفة بالبرمجيات منها ما يعمل بشكل ثابت ولا تأخذ بعين الاعتبار متغيرات الشبكة، ومنها ما يعمل بشكل ديناميكي متغيّر حسب بارامترات معينة.

تم في هذا البحث تقييم أداء أهم الخوارزميات المستخدمة لموازنة الحمل في الشبكات المعرفة بالبرمجيات كالخوارزمية العشوائية، خوارزمية Robin Round، خوارزمية Robin Round الموزونة، خوارزمية Robin Round الديناميكية الموزونة الأقل تأخيراً، خوارزمية الاتصالات الأقل، خوارزمية الاتصالات الأقل الموزونة وذلك لاختيار الخوارزمية ذات الأداء الأفضل والتي تستطيع التعامل مع كميات الطلبات المتفاوتة الواردة إلى المخدمات.

الكلمات المفتاحية: SDN، موازنة الحمل، مخدم، المحاكي Mininet

*أستاذ مساعد في قسم النظم والشبكات الحاسوبية – كلية الهندسة المعلوماتية – جامعة تشرين – اللاذقية – سوريا

**طالبة دراسات عليا (ماجستير) في قسم النظم والشبكات الحاسوبية –كلية الهندسة المعلوماتية–جامعة تشرين–اللاذقية–سوريا

مجلة جامعة طرطوس للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (5) العدد(5) 2021

Tartous University Journal for Research and Scientific Studies - engineering Sciences Series Vol. (5) No. (5) 2021

Performance evaluation of controllers in software-defined networks

Dr. Mohammed Sobih*
Eng. Iman Khalil**

(Received 4 / 2/ 2021 . Accepted 23 / 6 / 2021)

□ ABSTRACT □

Due to the rapid growth in network based services usage, networks have to handle massive amounts of traffic, and serve thousands of clients. Single server is unable to handle such a huge load. The solution is to use multiple servers with a load balancer acting as a front-end. The load balancer receives requests from multiple clients and depending on the different load balancing strategies it distribute these incoming requests among number of servers. Load balancing strategies in software-defined networks vary. Some operate statically and don't take into account network changes, and some operate dynamically according to certain parameters. In this paper, the performance of the most important algorithms used for load balancing in software-defined networks has been evaluated, such as Random algorithm, Round Robin algorithm, Weighted Round Robin algorithm, Least Delay Dynamic Weighted Round Robin algorithm ،Least Connections algorithm, Weighted Least Connections algorithm. The purpose is choosing the algorithm with the best performance which can deal with varying amounts of requests coming to the servers.

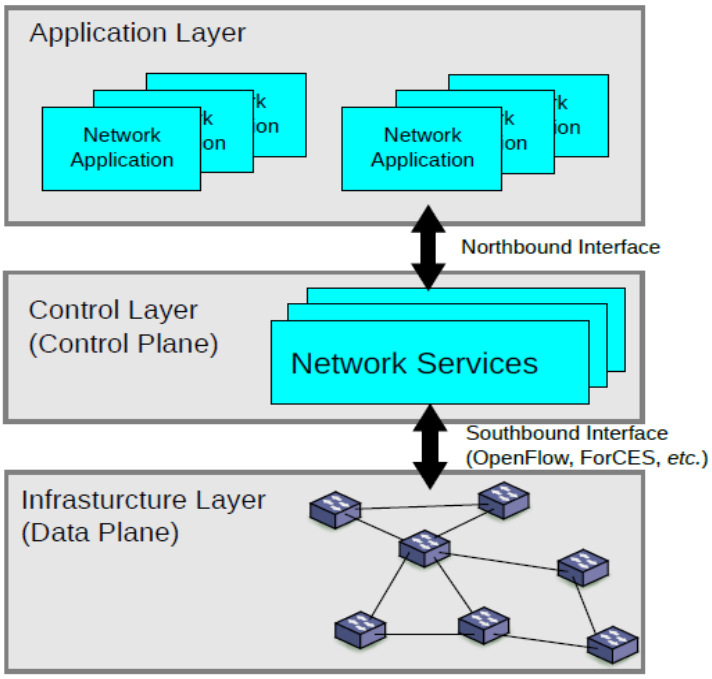
Keywords: SDN, Load balancing, server, Mininet simulator

* Professor, Department of Networks and Operating Systems. Faculty of Information Technology, Tishreen University, Lattakia, Syria

** Postgraduate Student (MSc), Department of Networks and Operating Systems. Faculty of Information Technology, Tishreen University, Lattakia, Syria

1- مقدمة

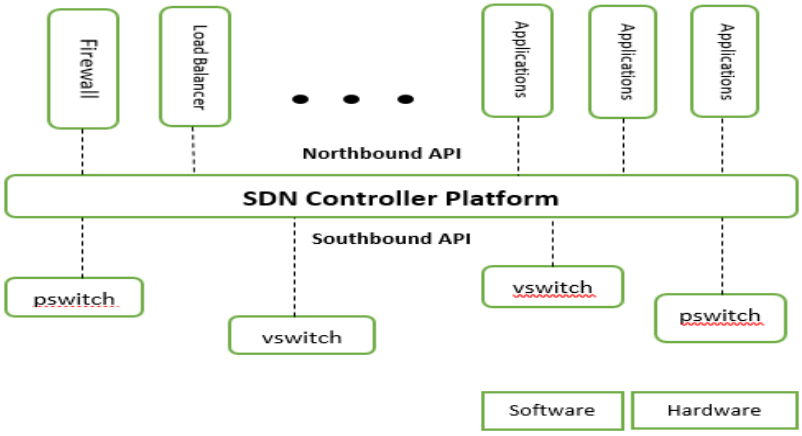
حظيت الشَبكات المعرَّفة بالبرمجيات بالكثير من الاهتمام في السنوات الأخيرة. فقد قدَّمت العديد من الميزات وساهمت في حل الكثير من المشاكل التي عانت منها الشبكات التقليدية. تعتمد الشبكات المعرفة بالبرمجيات على فصل عمليات توجيه البيانات عن عمليات التحكم و اتخاذ قرارات التوجيه لجعل التحكم بالشبكة مركزياً [1] كما يظهر في الشكل (1) .



الشكل (1) بنية شبكة SDN

يُعد بروتوكول OpenFlow الأكثر شيوعاً واستخداماً في متحكمات الشبكات المعرفة بالبرمجيات للتواصل والتحكم بالمبدلات. باستخدام هذا البروتوكول، يتعلم المبدل معلومات التوجيه من المتحكم ومن ثم يقوم بتمرير حزم البيانات بالاعتماد على هذه المعلومات [1] .

عرّفت منظمة ONF (Open Networking Foundation) البنية المعمارية لتقنية الشبكات المعرفة بالبرمجيات بنموذج مكون من ثلاث طبقات: طبقة التطبيقات، طبقة التحكم، وطبقة البنية التحتية [2]. تتكون طبقة التطبيقات من الخدمات والتطبيقات التي تقدمها الشبكة للمستخدم مثل التوجيه – QOS – ACL ويتواصل المتحكم مع هذه الطبقة عن طريق الواجهة الشمالية API Northbound كما يتواصل مع طبقة البنية التحتية عن طريق واجهة التخاطب الجنوبية Southbound API [2] كما في الشكل (2).



الشكل (2) واجهة التخاطب الشمالية والجنوبية

توفر الشبكات المعرفة بالبرمجيات (SDN) نهجاً فعالاً من حيث التكلفة ومرناً في تنفيذ موازنة الحمل، فهي لا تتطلب أي جهاز منفصل يعمل كموازن حمل. بل تُطَبَّق خوارزمية موازنة الحمل على المتحكم فيعمل المبدل كموازن حمل في الشبكة وبالتالي قللت من التكلفة. كما حلت الشبكات المعرفة بالبرمجيات مشكلة تقييد الشركات المصنعة كونها قابلة للبرمجة، فوفرت آلية فعالة ومرنة في تنفيذ موازنة الحمل. اذ تسمح للمبرمجين ببناء وتصميم الخوارزمية الخاصة بهم وتنفيذها بسهولة في الشبكة [3] .

بينما في الشبكات التقليدية تتم موازنة الحمل باستخدام أجهزة مخصصة لموازنة الحمل لكن هذه الاجهزة باهظة الكلفة وغير مرنة لأنها مغلقة من قبل الشركة المصنعة وغير مفتوحة المصدر وبالتالي غير قابلة للبرمجة من قبل مسؤولي الشبكة وليس لديهم القدرة على بناء الخوارزميات الخاصة بهم والمناسبة للشبكة . [3]

2- أهمية البحث و أهدافه

طُوِّرَت العديد من خوارزميات موازنة الحمل في الشبكات المعرفة بالبرمجيات، يُعد اختيار الخوارزمية المناسبة والتي تستطيع تلبية طلبات العملاء المتزايدة تحدياً بذاته، حيث يجب أن تحقق خوارزمية موازنة الحمل معدل انتاجية Throughput عالي وزمن استجابة منخفض، بالإضافة إلى توزيع الحمل بشكل متكافئ بين المخدمات بحيث لا يتم اغراق أحد المخدمات بالطلبات في حين يبقى مخدم آخر دون عمل.

تكمن أهمية البحث في اختيار خوارزميات فعالة لموازنة الحمل في الشبكات المعرفة بالبرمجيات وبالتالي زيادة كفاءة هذه الشبكات وتمكينها من تخديم أكبر عدد من طلبات المستخدمين بأقل تأخير.

إن الهدف من هذا البحث هو دراسة خوارزميات موازنة الحمل التالية: الخوارزمية العشوائية، خوارزمية Round Robin، خوارزمية Robin Round الموزونة، خوارزمية Robin Round الديناميكية الموزونة الأقل تأخيراً، خوارزمية الاتصالات الأقل، خوارزمية الاتصالات الأقل الموزونة. من ثم تقييم أدائها بالنسبة لعدة بارامترات هامة كعدد العمليات في الثانية وزمن الاستجابة، وذلك للوصول إلى اختيار الخوارزمية الأفضل.

3- طرائق البحث ومواده:

لتحقيقِ هدفِ البحث تمّ اتباعُ المنهَجيّة التالية:

- ❖ دراسة نظرية تناولت الجانبين التاليين:
  - o التعريف بالشبكات المعرفة بالبرمجيات وأهميتها.
  - o التعريف بخوارزميات موازنة الحمل المدروسة وشرح آلية عمل كل منها.
- ❖ الدراسة العملية وذلك من خلال استخدام المحاكي Mininet لبناء طوبولوجيا الشبكة وتنفيذ السيناريوهات المختلفة وذلك لمقارنة أداء الخوارزميات من حيث زمن الاستجابة وعدد العمليات في الثانية. كما قمنا باستخدام الأداة Httperf لقياس أداء مخدمات الويب، فهي توفر وسيلة مرنة لتوليد أحمال http مختلفة ومن ثم قياس استجابة المخدمات لهذه الأحمال.

3-1 الدراسة النظرية (الدراسات المرجعية):

الشبكات المعرفة بالبرمجيات هي تقنية ديناميكية قابلة للبرمجة وسهلة الإدارة، ذات هيكلية منظمة بشكل جيد تعتبر منهج جديد في إدارة شبكات الحاسب، حيث يستطيع مسؤول الشبكة إدارة الشبكة بطريقة مجرّدة بعيداً عن معرفة تفاصيل الشبكة في الطبقات السفلى [2]. تفصل الشبكة إلى مستويين:

- مستوى التحكم Control Plane
- مستوى البيانات Data Plane

حيث تقوم الشبكات المعرفة بالبرمجيات باستخلاص الوظائف الموجودة في مستوى التحكم من المبدلات Switches وتحقيقها بشكل برمجي ضمن تطبيقات مخصصة لهذه الغاية تسمى متحكمات Controllers.

إن المتحكم في الشبكات المعرفة بالبرمجيات مسؤول عن المحافظة على استقرار الشبكة وتطبيق السياسات والقواعد في الشبكة بالإضافة إلى توزيع التعليمات إلى أجهزة الشبكة المختلفة. فالمتحكم يملك نظرة عامة منطقية عن كل مكونات الشبكة [2].

كما أن نوع المتحكم ولغة البرمجة التي تم تطويره بها يؤثر بشكل كبير في أداء الشبكة [4]، كذلك فإن اختيار خوارزمية موازنة الحمل المناسبة يؤثر أيضاً على أداء الشبكة وقدرتها على معالجة أكبر عدد من الطلبات. تم تطوير العديد من خوارزميات موازنة الحمل منها ما يعمل بشكل ثابت ومنها ما يعمل بشكل ديناميكي. الخوارزميات الثابتة سهلة التنفيذ، لكنها ليست مرنة ولا تأخذ بالحسبان حالة المخدم (وحدة المعالجة المركزية، الذاكرة العشوائية ...). على سبيل المثال إذا تم تحميل أحد المخدمات بشكل زائد فإنه من الممكن ارسال الطلب التالي إلى نفس المخدم دون أخذ الحمل الإضافي بالحسبان. أما الخوارزميات الديناميكية فهي تأخذ بالحسبان حالة عقد الشبكة النشطة لكنها تهمل حجم ونوع طلب المستخدم [5]. قامت العديد من الأبحاث بدراسة أداء خوارزميات موازنة الحمل في شبكات SDN، ففي البحث [6] تمت مقارنة خوارزميتي Round Robin و Round Robin الموزونة في ظروف محددة للشبكة وقد تفوقت خوارزمية Round Robin الموزونة من حيث متوسط زمن الاستجابة وعدد العمليات في الثانية. في البحث [7] تم تقييم أداء خوارزمية Statistics التي تعتمد على اختيار المسار الأقل تحميل وتمت مقارنتها مع أداء خوارزميتي Round Robin و Round Robin الموزونة وقد تفوقت خوارزمية Statistics من حيث الإنتاجية وكانت الأقل تأخيراً.

قام الباحثون في البحث [8] بمقارنة الخوارزميات التالية Server Load ، , Flow Statistics Round Robin

وقد أظهر البحث تفوق خوارزمية Server Load من حيث الإنتاجية بينما كانت خوارزمية Flow Statistics هي الأفضل من حيث زمن الاستجابة. وفي البحث [9] تمت مقارنة خوارزميتي Round Robin والعشوائية من حيث الإنتاجية ومعدل ضياع الرزم، وذلك في ثلاثة سيناريوهات مختلفة حيث تم في السيناريو الأول اجراء المقارنة مع زيادة عدد المخدمات ثم مع زيادة عدد المستخدمين في السيناريو الثاني، وفي السيناريو الثالث تمت زيادة زمن المحاكاة، وقد تفوقت خوارزمية Round Robin على الخوارزمية العشوائية في السيناريوهات الثلاثة.

جميع هذه الأبحاث قامت بإجراء مقارنات محدودة من حيث عدد الخوارزميات المدروسة وظروف الشبكة المطبقة لذلك سنقوم في بحثنا هذا بمقارنة عدد من الخوارزميات مع بعضها البعض وضمن ظروف أكثر تنوعاً لتقييم أداء هذه الخوارزميات ضمن ظروف مختلفة ومدى تفاوت الأداء فيما بينها. سنقوم بدراسة الخوارزميات التالية:

3-1-1 الخوارزمية العشوائية Random algorithm

تعد هذه الخوارزمية ثابتة، حيث يختار المتحكم أحد المخدمات بشكل عشوائي باستخدام مولد رقم عشوائي، ولا يأخذ بالحسبان حالة المخدم أو الحمل الحالي عليه.

هذه الخوارزمية سهلة التنفيذ لكنها قد تسبب في بعض الأحيان توجيه حمل زائد إلى أحد المخدمات، بينما يكون مخدم آخر منخفض التحميل [10] .

3-1-2 خوارزمية Round Robin

هي خوارزمية بسيطة لموازنة الحمل، حيث يتم توزيع طلبات العملاء على المخدمات بترتيب دائري دون أي أولوية، عندما يصل إلى نهاية القائمة يكرر موازن الحمل التوزيع مرة أخرى، حيث يرسل الطلب التالي إلى المخدم الأول المدرج، ثم إلى المخدم الثاني، وهكذا [10]. وبالتالي إذا كانت **Si** العقدة التي تم اختيارها أخيراَ، يتم تعيين طلب جديد إلى العقدة

Si+1 ، حيث **i = (i + 1) mod N** و **N** هو عدد عقد المخدمات [3].

تقوم هذه الخوارزمية بتعيين حمل متساوٍ لكل عقدة ولا تأخذ بعين الاعتبار أي حالة من حركة المرور لذلك تصنف على أنها خوارزمية ثابتة. من مزايا خوارزمية Round Robin هي سهولة تنفيذها، فهي لا تحتاج إلى أجهزة بمواصفات فائقة، لا تحتاج إلى ذاكرة كبيرة ومواصفات عالية لوحدة المعالجة المركزية لكي يتم تنفيذها [3] .

3-1-3 خوارزمية Round Robin الموزونة

يتلقى كل مخدم وفق هذه الخوارزمية الطلبات من العملاء بناءً على المعايير التي يحددها مدير الشبكة. يتم تعيين وزن ثابت لكل مخدم، وهي قيمة عدد صحيح تشير إلى قدرة المعالجة. يحدد مدير الشبكة هذه الأوزان بما يتناسب مع القدرات الفعلية للمخدمات بحيث يتعامل المخدم الأقل قدرة مع كمية اقل من

الطلبات الواردة. على سبيل المثال، إذا كانت سعة المخدم 1 أكبر بخمس مرات من سعة المخدم 2، فيمكننا تعيين وزن قدره 5 للمخدم 1 ووزن 1 للمخدم 2.

تستطيع هذه الخوارزمية التمييز بين المخدمات غير المتجانسة، لكنها لا تحتاج إلى حساب عدد الاتصالات لكل عقدة، ولا تأخذ بعين الاعتبار التغيرات الأخرى الحاصلة في الشبكة لذلك تعتبر خوارزمية ثابتة [11].

3-1-4 خوارزمية Round Robin الموزونة الأقل تأخيراً

تعد هذه الخوارزمية مشابهة لخوارزمية Round Robin الموزونة، لكن هنا يُعَيّن وزن لكل مخدم بناءً على سرعة الوصلة بين المخدم والمبدل بحيث يُعطى المخدم ذو الاتصال الاسرع وزن أكبر ليتعامل مع عدد أكبر من الطلبات. على سبيل المثال، اذا كان هناك تأخير في الوصلة بين المخدم 1 والمبدل قدره 20 ميلي ثانية، ويوجد تأخير بين المخدم 2 والمبدل 10 ميلي ثانية، في هذه الحالة سيقوم مدير الشبكة بتعيين وزن 2 للمخدم 2 ووزن 1 للمخدم 1 وبالتالي سيتم توجيه عدد أكبر من الطلبات إلى المخدم 2 [12] .

3-1-5 خوارزمية الاتصالات الأقل (Least-Connections algorithm)

تقوم هذه الخوارزمية بتوجيه الطلبات المستلمة من الشبكة إلى العقدة ذات أقل عدد من الاتصال المتزامنة. وهي إحدى الخوارزميات الديناميكية؛ لأنها تحتاج إلى تتبع العدد الإجمالي للاتصالات النشطة المتزامنة لكل عقدة. تأخذ هذه الخوارزمية بعين الاعتبار الحمل الحالي على المخدم.

وفق هذه الخوارزمية يقوم المتحكم كل خمس ثواني بإرسال رسالة stats-request إلى المبدلات للاستعلام عن محتوى جدول التدفق flow table، وعندما يستقبل المتحكم رسالة ال stats-reply يقوم بحساب عدد الاتصالات الحالية المرتبطة بكل مخدم ويقارنها مع بعضها ليرسل الطلب التالي للمخدم ذي عدد الاتصالات الاقل [13].

3-1-6 خوارزمية الاتصالات الأقل الموزونة(Weighted Least-Connections algorithm)

هذه الخوارزمية مبنية على خوارزمية Least-Connections algorithm لكنها تأخذ بعين الاعتبار أيضاً اختلاف مواصفات المخدمات، حيث يمكن لمسؤول الشبكة تعيين وزن حسب أداء كل مخدم ، و ستتلقى المخدمات ذات الوزن الأعلى نسبة مئوية أكبر من الاتصالات المتزامنة [14].

تعمل هذه الخوارزمية على النحو التالي:

على افتراض لدينا S عقدة مخدم حيث $S = \{S_0, S_1, \ldots, S_{n-1}\}$، $W(S_i)$ تمثل وزن العقدة S_i، $C(S_i)$ تمثل عدد الاتصالات الحالية للعقدة S_i .

فيكون مجموع أعداد الاتصالات لجميع المخدمات هو $C_{SUM} = \Sigma C(S_i)(i = 0,1,\ldots, n-1)$

وبالتالي يتم ارسال الطلب الجديد إلى المخدم S_m ، حيث S_m يحقق الشرط :

$$\frac{C(S_m)/C_{SUM}}{W(S_m)} = \min\left\{\frac{C(S_i)/C_{SUM}}{W(S_i)}\right\}$$

$$i = 0,1,\ldots, n-1, \quad W(S_i) \neq 0$$

المعادلة (1) : شرط اختيار المخدم التالي

وبما أن C_{SUM} ثابت في هذه العلاقة ، فيمكن تبسيطها كالتالي [14] :

$$C(S_m)/W(S_m) = \min\{C(S_i)/W(S_i)\}$$
$$i = 0,1,\ldots, n-1, \quad W(S_i) \neq 0$$

المعادلة (2): المعادلة المختصرة لاختيار المخدم التالي

3-1-7 أداة البحث

تم استخدام المحاكي [15] Mininet وهو محاكي يُنشئ شبكة من المضيفات الظاهرية، والمبدلات، والمتحكمات، والروابط. تستطيع مضيفات Mininet تشغيل برامج نظام Linux القياسية، كما تدعم مبدلاته البروتوكول OpenFlow لتحقيق توجيه مخصص للطرود عالي المرونة. يدعم Mininet البحث والتطوير والتعلم والنماذج الأولية والاختبار وتصحيح الأخطاء وأي مهام أخرى من خلال تقديم شبكة كاملة على كمبيوتر محمول أو أي جهاز آخر.

يعاني هذا المحاكي من قيود إذ لا يمكن تشغيل مبدلات وتطبيقات OpenFlow غير المتوافقة مع نظام Linux.

كما تم استخدام الأداة Httperf [16] لقياس أداء مخدمات الويب، فهي توفر وسيلة مرنة لتوليد أحمال Http مختلفة ومن ثم قياس استجابة المخدمات لهذه الأحمال.

3-2 الدراسة العملية:

في البداية تم تقييم أداء الخوارزميات الستة المدروسة من حيث عدد العمليات في الثانية وزمن الاستجابة وذلك في شبكة متماثلة من حيث المخدمات، حيث تم زيادة معدل الطلبات المرسلة من المضيفين تدريجياً وذلك باستخدام الأداة Httperf والتي تقوم بتوليد طلبات http وترسلها إلى المتحكم، ليقوم بدوره بتطبيق خوارزمية موازنة الحمل واختيار المخدم المناسب وفقاً لآلية عمل الخوارزمية.

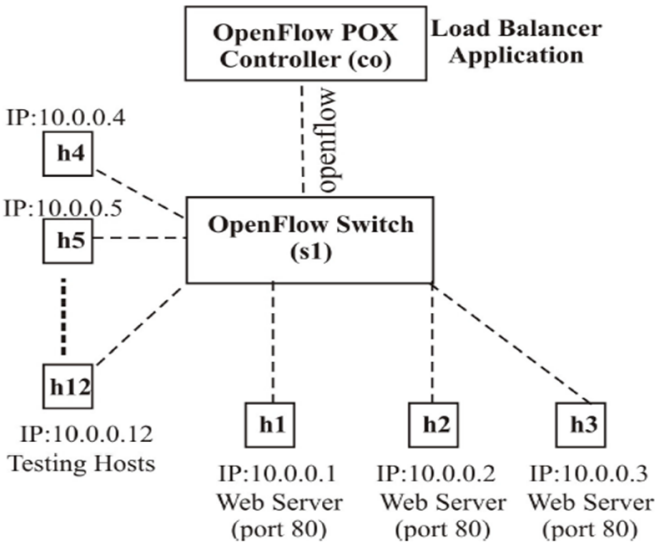
ثم تم قياس زمن الاستجابة وعدد العمليات في الثانية بعد بتغيير ظروف الشبكة وذلك بإضافة أزمنة تأخير مختلفة على الوصلات بين المبدل والمخدمات. بعد ذلك تم قياس هذه البارامترات بعد تغيير مواصفات المخدمات وتحديداً سرعة المعالجة فيها، وفي السيناريو الأخير تم تقييم أداء الخوارزميات مع وجود حمل إضافي موجه إلى أحد المخدمات فقط. كل هذه السيناريوهات تم تطبيقها من خلال بناء شبكة مؤلفة من متحكم و مبدل و مضيفَين متصلين به باستخدام المحاكي Mininet. تم القيام بالتجارب على جهاز كمبيوتر بنظام Ubuntu و بمعالج Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz (4 CPUs), ~2.4GHz و ذواكر 4 غيغابايت.

4- النتائج و المناقشة:

تم تنفيذ عدة سيناريوهات

السيناريو الأول: في البداية تم انشاء شبكة مؤلفة من متحكم واحد من نوع pox و مبدل واحد و ثلاثة مخدمات ويب متماثلة من حيث المواصفات، عرض الحزمة المطبق بين المخدمات والمبدل 20 Mbps

كما في الشكل (3) :



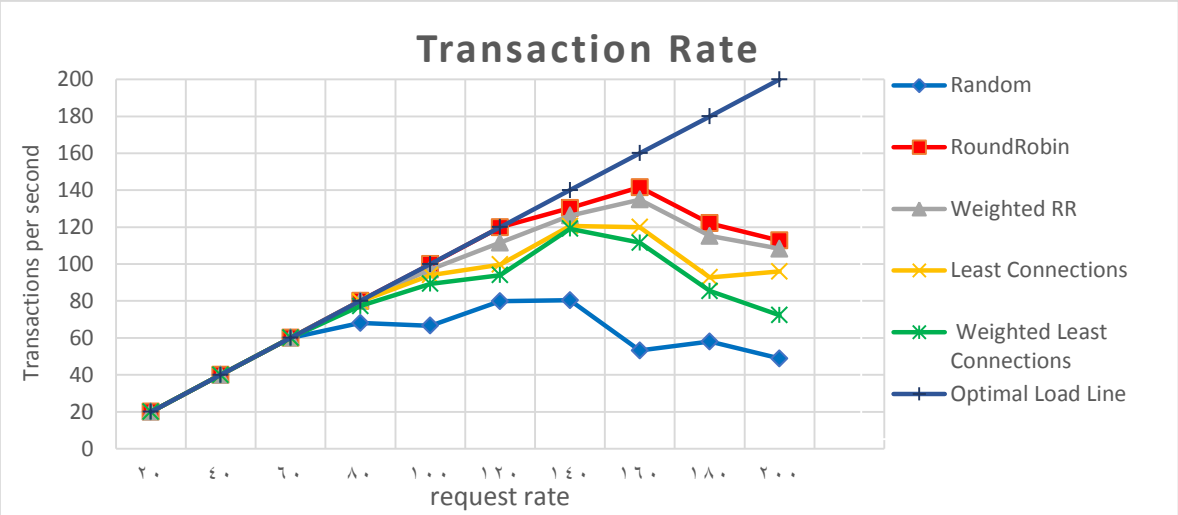
الشكل (3) : طبولوجية الشبكة المستخدمة

وباستخدام الأداة httperfتم توليد حمل من طلبات http المتتالية بحجم ثابت إلى المخدمات بزمن محاكاة 100 sec. وتم زيادة عدد الطلبات في الثانية تدريجياً بدءاً من 20 طلب في الثانية وصولاً إلى 200 طلب في الثانية باستخدام الأمر التالي:

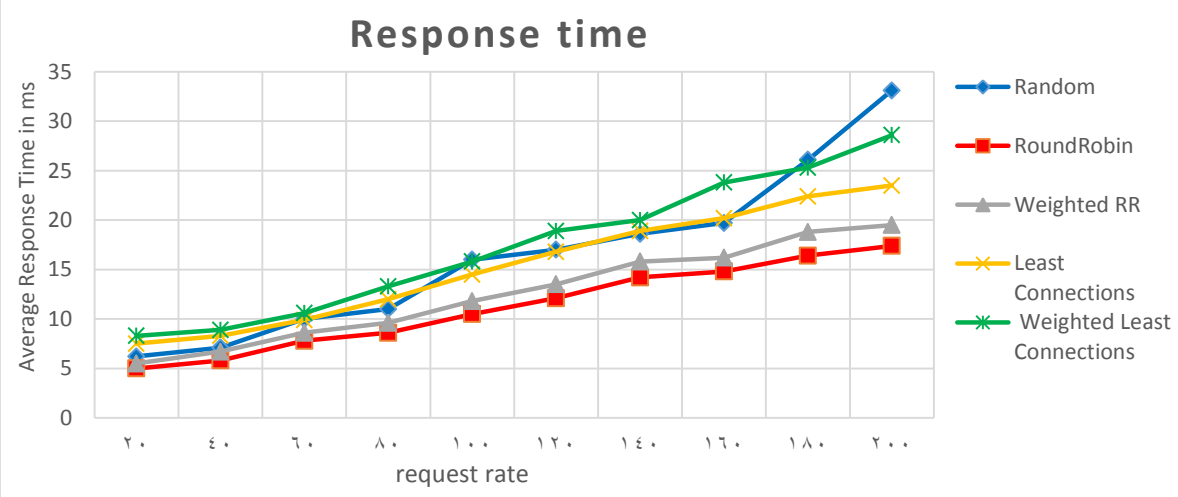
httperf --server 10.0.1.1 --port 80 --num-conns 2000 --rate 20 --uri /index.html --timeout 5

حيث server: عنوان IP المخدم المراد ارسال الطلبات إليه، port: رقم المنفذ الذي يعمل عليه المخدم، num-conns : إجمالي عدد الاتصالات المراد إنشاؤها ، rate : عدد الطلبات في الثانية ، uri : مسار صفحة الويب المراد تنزيلها ، timeout : زمن الانتظار الأقصى للاستجابة للطلب .

أظهرت النتائج أن تفوق خوارزمية Round Robin على بقية الخوارزميات وذلك لأن منحني الخوارزمية هو الأقرب من خط الحمل المثالي والذي يعبر عن عدد العمليات التي تنفذها المخدمات في الثانية في الحالة المثالية لتوزيع الحمل كما في الشكل (4)، كما تفوقت من حيث زمن الاستجابة الشكل (5) .



الشكل (4) عدد العمليات في الثانية في الخوارزميات المدروسة مع زيادة معدل الطلبات الواردة



الشكل (5) متوسط زمن الاستجابة في الخوارزميات المدروسة مع زيادة معدل الطلبات الواردة

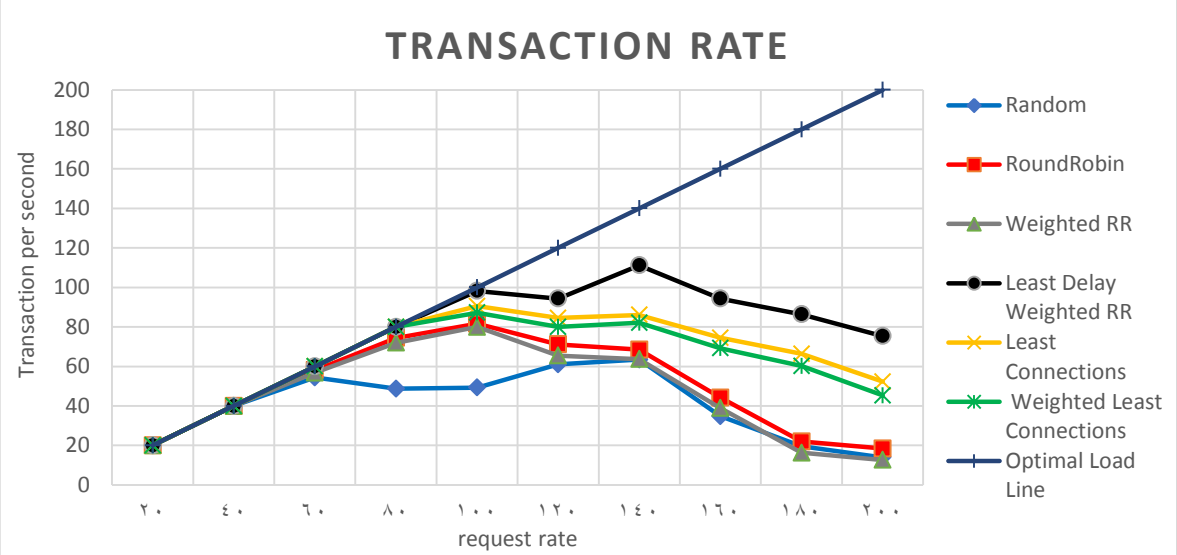
إن سبب الأداء الجيد لخوارزمية Round Robinهو قدرتها على توزيع الحمل بشكل موحد على جميع المخدّمات المتماثلة، بالإضافة إلى بساطة هذه الخوارزمية وسهولة تنفيذها، مما أعطاها تفوق على باقي الخوارزميات كخوارزميتي Least-Connections algorithm و Weighted Least-Connections algorithmاللتين تحتاجان إلى زمن إضافي للتنفيذ.

السيناريو الثاني: قمنا بإنشاء شبكة مؤلفة من متحكم واحد من نوع pox ومبدل واحد وثلاثة مخدمات متماثلة من حيث المواصفات تعمل كمخدمات ويب، عرض الحزمة المطبق بين المخدمات والمبدل 20 Mbps

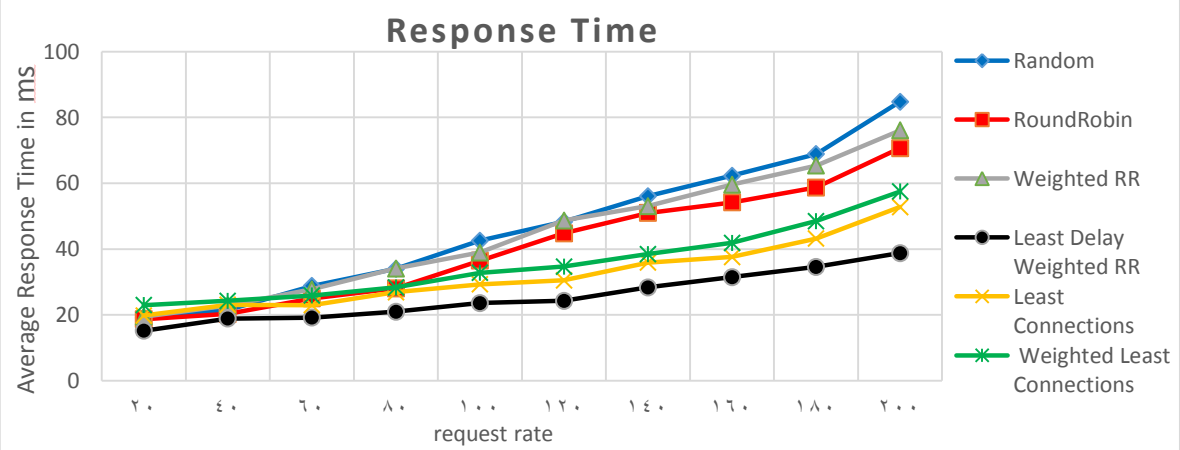
وقمنا بإضافة تأخير على الوصلات بين المخدمات والمبدل كالتالي:

5 ms بين المخدم الأول والمبدل ، 10 ms بين المخدم الثاني والمبدل ، 20 ms بين المخدم الثالث والمبدل

وباستخدام الأداة httperf قمنا بتوليد حمل من طلبات http المتتالية بحجم ثابت إلى المخدمات بزمن محاكاة 100 sec . وتم زيادة عدد الطلبات في الثانية تدريجياً بدءاً من 20 طلب في الثانية وصولاً إلى 200 طلب في الثانية. أظهرت النتائج تفوق خوارزمية Round Robin الموزونة الأقل تأخيراً على باقي الخوارزميات وذلك من حيث عدد العمليات في الثانية حيث أن منحني الخوارزمية هو الأقرب من منحني الحمل المثالي الشكل (6) ، و زمن الاستجابة الشكل (7) .



الشكل (6) عدد العمليات في الثانية في الخوارزميات المدروسة مع زيادة معدل الطلبات الواردة



الشكل (7) متوسط زمن الاستجابة في الخوارزميات المدروسة مع زيادة معدل الطلبات الواردة

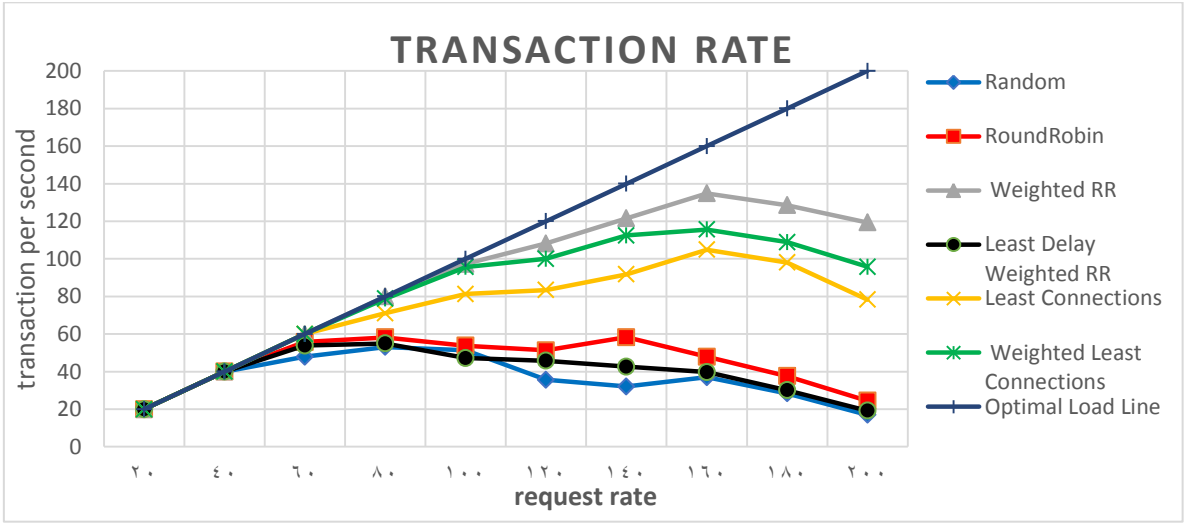
نلاحظ أن خوارزمية Round Robin الموزونة الأقل تأخيراً قد تفوقت على باقي الخوارزميات، وذلك لأن هذه الخوارزمية تقوم بتعيين الأوزان للمخدمات بناءً على سرعة الوصلة بين المخدم والمبدل بحيث يُعطى المخدم ذو الاتصال الأسرع وزن أكبر، بالإضافة إلى كونها خوارزمية ثابتة لا يتطلب تنفيذها زمناً كبيراً بالمقارنة مع الخوارزميات الديناميكية.

السيناريو الثالث: قمنا بإنشاء شبكة مؤلفة من متحكم واحد من نوع pox ومبدل واحد وثلاثة مخدمات مختلفة من حيث المواصفات تعمل كمخدمات ويب كالتالي:

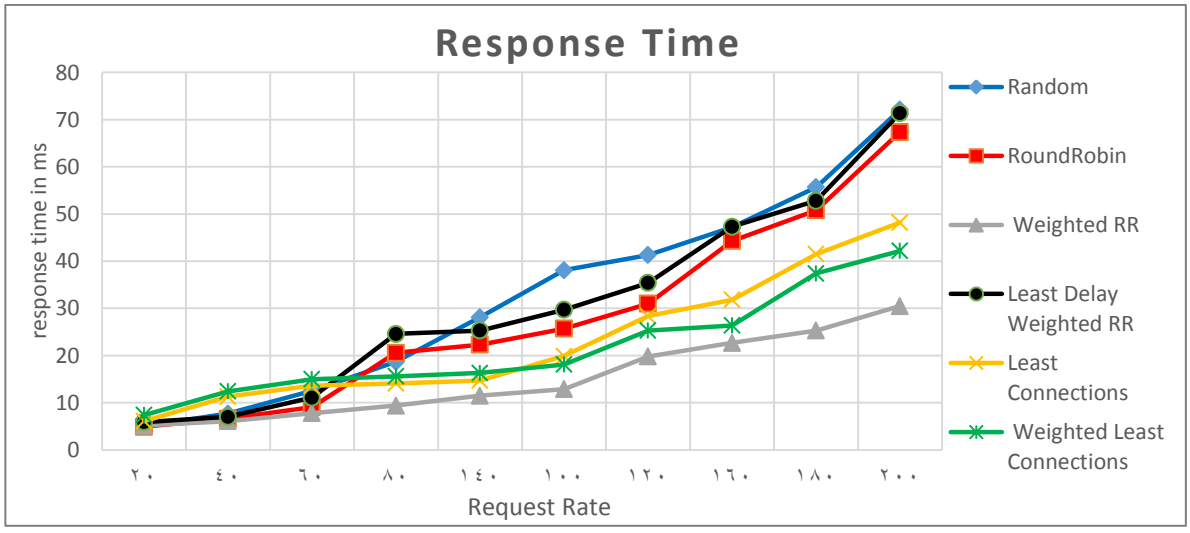
المخدم الأول يعمل بقدرة 30% من موارد الجهاز ، المخدم الثاني يعمل بقدرة 60% من موارد الجهاز ،المخدم الثالث يعمل بقدرة 10% من موارد الجهاز ، وعرض الحزمة المطبق بين المخدمات والمبدل 20 Mbps.

باستخدام الأداة httperf قمنا بتوليد حمل من طلبات http المتتالية بنفس الحجم إلى المخدمات بزمن محاكاة 100 sec. وتم زيادة عدد الطلبات في الثانية تدريجياً بدءاً من 20 طلب في الثانية وصولاً إلى 200 طلب في الثانية .

أظهرت النتائج تفوق خوارزمية Weighted Round Robin على باقي الخوارزميات وذلك من حيث عدد العمليات في الثانية الشكل (8) ، و زمن الاستجابة الشكل (9) .



الشكل (8) عدد العمليات في الثانية في الخوارزميات المدروسة مع زيادة معدل الطلبات الواردة



الشكل (9) متوسط زمن الاستجابة في الخوارزميات المدروسة مع زيادة معدل الطلبات الواردة

نلاحظ أن خوارزمية Weighted RR قد تفوقت على باقي الخوارزميات وذلك لأن هذه الخوارزمية تقوم بتعيين الأوزان للمخدمات بناءً على مواصفاتها بحيث يتعامل المخدم الأقل قدرة مع كمية أقل من الطلبات الواردة. كما تمكنت خوارزمية Weighted Least-Connections algorithm من العمل بشكل جيد لأنها

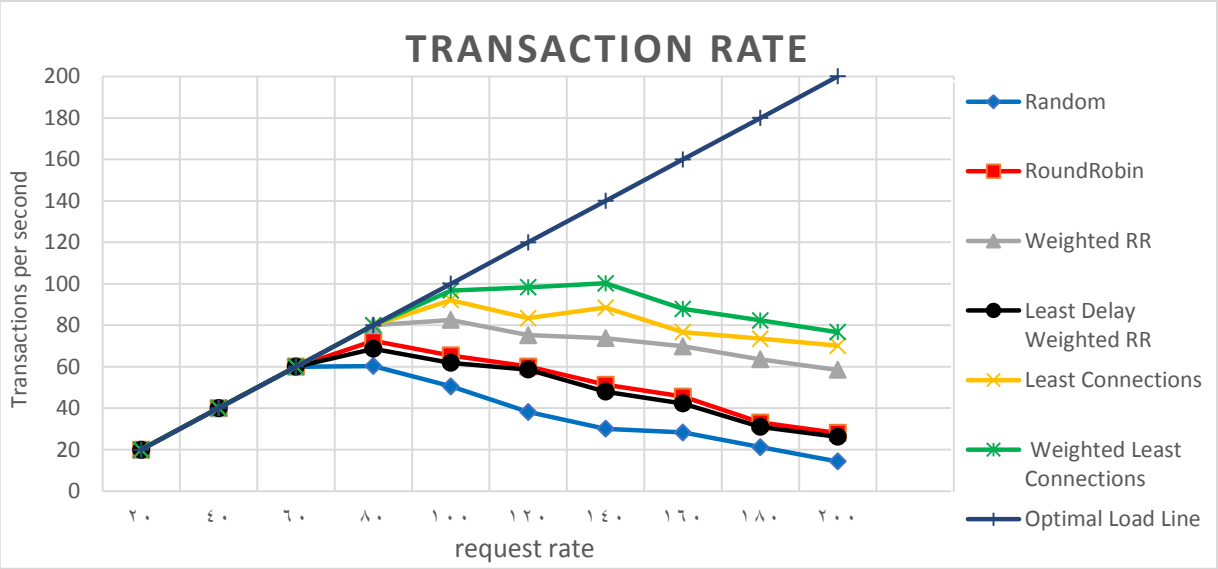
تأخذ بعين الاعتبار اختلاف قدرة المخدمات على المعالجة لكن تنفيذ هذه الخوارزمية يتطلب زمن إضافي بالمقارنة مع خوارزمية Weighted RR الثابتة مما أعطى الخوارزمية الثابتة تفوقاً عليها.

السيناريو الرابع: تم انشاء شبكة مؤلفة من متحكم واحد من نوع pox و مبدل واحد و ثلاثة مخدمات مختلفة من حيث المواصفات تعمل كمخدمات ويب كالتالي :

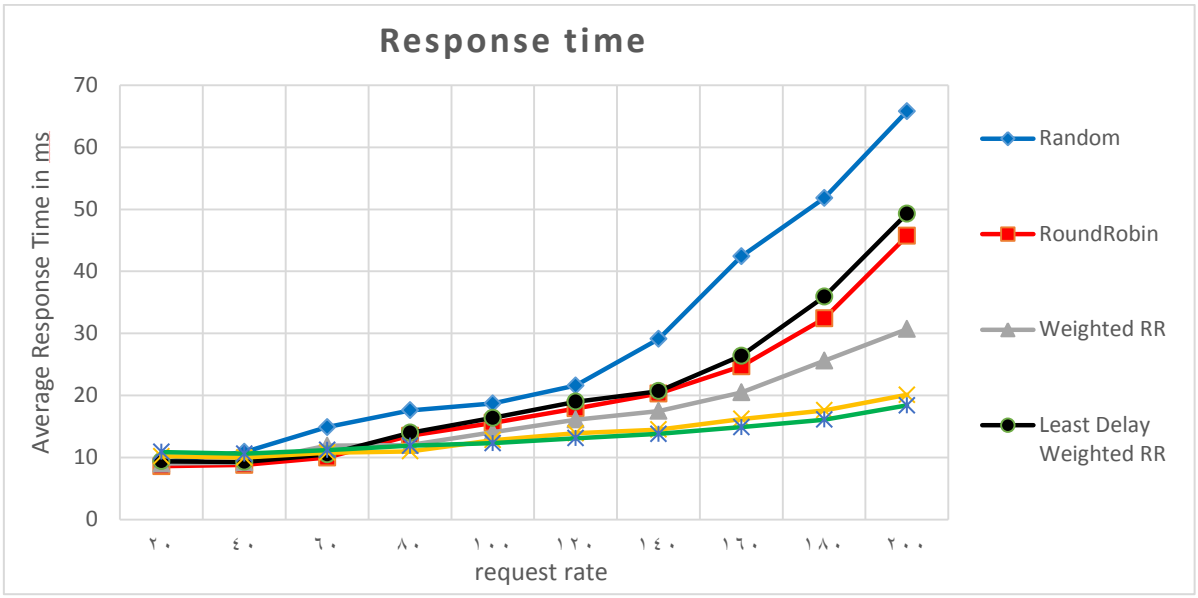
المخدم الأول يعمل بقدرة 60% من موارد الجهاز ، المخدم الثاني يعمل بقدرة 30% من موارد الجهاز ،المخدم الثالث يعمل بقدرة 10% من موارد الجهاز ، وعرض الحزمة المطبق بين المخدمات والمبدل 20 Mbps.

وباستخدام الأداة httperf تم توليد حمل من طلبات http المتتالية بنفس الحجم إلى المخدمات بزمن محاكاة 100 ثانية وتم زيادة عدد الطلبات في الثانية تدريجياً بدءاً من 20 طلب في الثانية وصولاً إلى 200 طلب في الثانية ، و بالتزامن تم توجيه حمل مستقل من الطلبات إلى المخدم 2 فقط وذلك لاختبار أداء الخوارزميات مع وجود حمل متغير على أحد المخدمات .

أظهرت النتائج تفوق خوارزمية Least-Connections algorithm على باقي الخوارزميات وذلك من حيث عدد العمليات في الثانية الشكل (10) ، و زمن الاستجابة الشكل (11) .



الشكل (10) عدد العمليات في الثانية في الخوارزميات المدروسة مع زيادة معدل الطلبات الواردة



الشكل (11) متوسط زمن الاستجابة في الخوارزميات المدروسة مع زيادة معدل الطلبات الواردة

في هذا السيناريو تفوقت خوارزمية Weighted Least Connections على باقي الخوارزميات وذلك لأنها خوارزمية ديناميكية قادرة على التأقلم مع العوامل المتغيرة التي تطرأ على الشبكة بالإضافة إلى كونها تقوم بتعيين أوزان للمخدمات حسب قدرتها، كما أن خوارزمي Least Connections قد عملت بشكل جيد لأنها خوارزمية ديناميكية أيضاً لكنها لا تأخذ بعين الاعتبار اختلاف مواصفات المخدمات. بينما لم تتمكن خوارزمية Weighted RR من التكيف مع وجود حمل متغير في الشبكة على الرغم من أنها تراعي اختلاف مواصفات المخدمات وذلك لأنها خوارزمية ثابتة.

5- الاستنتاجات :

تم في هذا البحثِ تقييمِ أداءِ ست خوارزميات الخوارزمية العشوائية، خوارزمية Round Robin، خوارزمية Robin Round الموزونة ، خوارزمية Robin Round الديناميكية الموزونة الأقل تأخيراً ، خوارزمية الاتصالات الأقل ، خوارزمية الاتصالات الأقل الموزونة ، من خلالِ بارامترين هما عدد العمليات في الثانية ، و زمن الاستجابة وذلك في ظروف مختلفة للشبكة و المخدمات .

بناءً على النتائج التي تم التوصل إليها في البحث يمكن استنتاج مايلي:

- زمن الاستجابة في الخوارزميات الثابتة أقل من زمن الاستجابة في الخوارزميات الديناميكية عندما يكون عدد الطلبات قليل في الشبكة.
- عندما تكون ظروف الشبكة ثابتة ومحددة مسبقاً تستطيع الخوارزميات الثابتة التي تأخذ هذه الظروف بعين الاعتبار (Round Robin , Least Delay Weighted Round Robin Weighted Round Robin,) أن تتفوق على الخوارزميتين الديناميكيتين .

- الخوارزمية العشوائية لم تتمكن من إعطاء أداء جيد في أي حالة من الحالات عندما يكون عدد الطلبات في الثانية كبير، في حين أنها أعطت أداء جيد نسبياً عندما يكون عدد الطلبات في الثانية قليل (أقل من 60 طلب في السيناريوهات المطبقة) .
- تستطيع الخوارزميتين الديناميكيتين (Least , Weighted Least Connections Connections) أن تتكيفا مع مختلف ظروف الشبكة وتعطيا أداء جيد نسبياً .
- تتفوق الخوارزميتين الديناميكيتن (Least , Weighted Least Connections Connections) على باقي الخوارزميات الثابتة عندما تكون ظروف الشبكة متغيرة .

6- التوصيات:

نوصي بناءً على ما سبق باستخدام الخوارزميات الثابتة عندما تكون ظروف الشبكة ثابتة ولا يوجد فيها عوامل متغيرة

(كوجود تأخير معروف مسبقاً غير متغير مع الوقت على الوصلات بين المبدل والمخدمات) حيث تتفوق في هذه الحالة الخوارزمية الثابتة التي تأخذ ظروف الشبكة بعين الاعتبار. أما في حال كانت ظروف الشبكة متغيرة (كوجود حمل متغير على المخدمات) فإن استخدام احدى الخوارزميتين الديناميكيتين يكون أفضل.

7- المراجع References

[1] Braun,W ; Menth,M . 2014, *Software-defined networking using OpenFlow: Protocols, applications and architectural design choices. Future Internet*. *Future Internet* .Vol.6(2) , 302-336.
[2] Bailey,S ; Bansal,D ; Dunbar,L ; Hood,D ; Maguire,J ; Malek,D ; Meyer,D ; Paul,M ; Schaller,S ; Schneider,F ; Sherwood,R ; Varma,E . 2013 , *SDN Architecture Overview. Version 1.0 ,Open Networking foundation ONF*
[3] Mithbavkar,D ; Joshi,H ; Kotak,H ; Gajjar,D ; Perigo,L . 2016 , *Round robin load balancer using software defined networking (SDN). Capstone Team Research Project, 5*,1-9
[4] Sobih,M ; Abo Obaid,A . 2018 , *Performance evaluation of controllers in software-defined networks. Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series*, Vol. (04) No. (5) 8408
[5] Al-Mashhadi,S ; Anbar,M; Jalal,R.,A ; Al-Ani, A. 2020, *Design of cloud computing load balance system based on SDN technology. In Computational Science and Technology* Springer, Singapore ,(pp. 123-133)
[6] Ikram,A ; Arif,,S ; Ayub,N ; Arif,W . 2018, *Load Balancing In Software Defined Networking (SDN). MAGNT Research Report*, Vol.5(1). PP. 298-305
[7] Fancy,C ; Pushpalatha, M ; Pushpa . 2019, *Experimentation of Traditional Load Balancing Algorithms in Software Defined Network. International Journal of Recent Technology and Engineering (IJRTE)* , Vol.8 ,2277-3878
[8] Joshi , N ; Gupta, D . 2019 , *A Comparative Study on Load Balancing Algorithms in Software Defined Networking. International Conference on Ubiquitous Communications and Network Computing*, Springer, Cham, pp. 142-150
[9] Jadhav,K,A ; Mulla,M.M ; Narayan,D,G . 2020, *An Efficient Load Balancing Mechanism in Software Defined Networks . 12th International Conference on Computational Intelligence and Communication Networks (CICN)* IEEE , pp. 116-122
[10] Kaur,S ; Kumar,K ; Singh,J ; Ghumman,N,S. 2015, *Round-robin based load balancing in Software Defined Networking. In 2nd International Conference on Computing for Sustainable Global Development (INDIACom)* IEEE . pp. 2136-2139
[11] Sabiya,J,S. 2016 , *Weighted round-robin load balancing using software defined networking. International Journal of Advanced Research in Computer Science and Software Engineering, 6*(6), 621-625.
[12] Sroya,M,S ; Singh,V. 2017 , *LDDWRR: Least Delay Dynamic Weighted Round-Robin Load Balancing in Software Defined Networking. International Journal of Advanced Research in Computer Science* .Vol.8(5) , pp.145-148
[13] Zhang, H ; Guo, X. 2014, *SDN-based load balancing strategy for server cluster.In IEEE 3rd International Conference on Cloud Computing and Intelligence Systems*.pp.662-667
[14] Ren,X ; Lin,R ; Zou,H . 2011, *A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast. In 2011 IEEE international conference on cloud computing and intelligence systems,* pp. 220-224
[15] http://mininet.org , last visited at December 30, 2020
[16] https://github.com/httperf/httperf , last visited at December 22, 2020