

تحسين أداء الشبكات المعرفة برمجياً ذات

المتحكمات المتعددة من خلال خوارزمية

موازنة الحمل

د. أحمد صقر أحمد * حازم ديب **

ملخص البحث:

تم اقتراح الشبكات المعرفة برمجياً (SDN) Software Defined Network كحل لمعالجة المشاكل التي تعاني منها الشبكات التقليدية، ولكن النموذج الأولي المقترح لشبكات SDN، والذي يعتمد على وجود متحكم مركزي وحيد، عانى من مشكلة رئيسية هي مشكلة نقطة الفشل الوحيدة، هذا الأمر دفعنا إلى الانتقال إلى بنية شبكات SDN بمتحكمات متعددة، وقد واجهت هذه البنية الجديدة تحديات عديدة أهمها هي عملية موازنة الحمل بين المتحكمات المتعددة الموجودة في الشبكة، بحيث لا يتعرض أحد هذه المتحكمات إلى حمل زائد بشكل يجعله يتوقف عن العمل، لذا فقد كان لعملية اختيار خوارزمية موازنة حمل مناسبة دور كبير في أداء الشبكة.

تم التطرق في هذا البحث إلى الحاجة إلى الشبكات المعرفة برمجياً وبنيتها، والحاجة إلى الانتقال من البنية المركزية إلى البنية الموزعة، مع ذكر فوائد البنية الموزعة والتحديات التي تتعرض لها، والنماذج المختلفة منها، كما تم أيضاً تعريف عملية موازنة الحمل، والتعرف على آليتها، والعوامل التي تلعب دوراً في هذه العملية، بالإضافة إلى أنه تم ذكر أهم خوارزميات موازنة الحمل المستخدمة في وقتنا الحالي.

وفي الجانب العملي، تم اقتراح خوارزمية جديدة لموازنة الحمل وقد تمت تسميتها بخوارزمية الترحيل باستخدام العتبات الثلاث، حيث تمت مقارنة هذه الخوارزمية مع مجموعة من خوارزميات موازنة الحمل المستخدمة سابقاً، من خلال ثلاثة عوامل هي: إنتاجية الشبكة، معدل استخدام الموارد، واحتمال فشل المتحكم، وقد بينت نتائج التجارب التي تم إجراؤها أن الخوارزمية المقترحة تعطي نتائج أفضل من الخوارزميات السابقة،

وتقدم تحسناً مقبولاً على جميع الخوارزميات الأخرى وفقاً لجميع العوامل والسيناريوهات التي تمت دراستها في هذا البحث.

الكلمات المفتاحية: الشبكات المعرفة برمجياً، موازنة الحمل، إنتاجية الشبكة، معدل استخدام الموارد.

*أستاذ-قسم النظم والشبكات الحاسوبية-كلية الهندسة المعلوماتية-جامعة تشرين-اللاذقية-سوريا.

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

**طالب دكتوراه-قسم النظم والشبكات الحاسوبية-كلية الهندسة المعلوماتية-جامعة تشرين-اللاذقية-

سوريا. Email: hazem.deeb@tishreen.edu.sy

Improvement of Multiple Software Defined Networks Controllers with Load Balancing Algorithm

***Dr. Ahmad Saker Ahmad**

****Hazem Deeb**

Abstract

Software Defined Networks were proposed as a solution for the problems that traditional networks suffer from. But the first proposed SDN prototype which depends on having a single main controller suffered from a main problem of having single point of failure, which is what force us to go with multiple controllers SDN, this new architecture faced multiple challenges such as load balancing between the different controllers in the network, so that no controller gets overloaded to the point which makes it stop working. So choosing the correct load balancing algorithm had a major rule in the performance of the network.

This research discussed the need for software defined networks, their infrastructures and the need to move from the central infrastructure to the distributed one with mentioning the benefits of distributed infrastructures, the challenges that it's facing and the different types that it takes. The load balancing process, its mechanism and the factors that play an important role in this process were introduced. In addition to that, the most common load balancing algorithms used nowadays were also mentioned in this research.

A new algorithm was proposed to balance the load, which was called "Three thresholds migration algorithm", this algorithm was compared to multiple previously used load balancing

algorithms by using three factors: network throughput, resource utilization, and controller failure probability. The experiments results show decent improvement over the other algorithms based on the different factors and scenarios studied in this research.

Keywords: SDN, Load balancing, Throughput, Resource Utilization.

***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

1- مقدمة:

كانت الفكرة الأساسية من شبكات SDN هي معالجة المشكلات التي تعاني منها الشبكات التقليدية، فقد اتسمت الشبكات التقليدية بقلّة المرونة وعدم القدرة على التكيف عند ازدياد حجم الشبكة بشكل كبير، وذلك لأنها تعتمد على عملية إدخال الإعدادات يدوياً لكل جهاز شبكي موجود في الشبكة، وعند حصول تعديل على بنية الشبكة مثل إضافة جهاز جديد وجب إدخال كم كبير من الإعدادات، وهذا يعتبر عملية معقدة للغاية وخاصةً في الشبكات الكبيرة، لذا تم اقتراح الانتقال إلى شبكات SDN والتي تهدف إلى جعل الشبكات أكثر مرونة، وتسهيل عملية الإدارة بفصل مستوي التحكم عن مستوي البيانات، وجعل مستوي التحكم موجود في جهاز واحد هو متحكم SDN المركزي الموجود في الشبكة. بدأت بنية شبكات SDN بوجود عدد من الأجهزة الشبكية مع متحكم SDN مركزي وحيد للشبكة ككل، وتتم جميع عمليات التحكم واتخاذ القرارات في هذا المتحكم، ولكن مع مرور الوقت وزيادة حجم الشبكات أصبح هذا المتحكم الوحيد يتعرض لحمل زائد، مما قد يسبب في توقفه عن العمل، وبما أنه لا يوجد بديل لهذا المتحكم سوف تظهر مشكلة نقطة الفشل الوحيدة في هذا النوع من الشبكات، وقد كان هذا الدافع الرئيسي للانتقال إلى بنية شبكات SDN موزعة بمتحكمات متعددة، حيث أتاحت هذه البنية تجنب المشاكل الحاصلة في الشبكات ذات المتحكم الوحيد.

2- أهمية البحث وأهدافه:

لقد برزت الحاجة الملحة للانتقال من بنية شبكات SDN ذات المتحكم المركزي إلى بنية ذات متحكمات متعددة، وذلك لمعالجة المشاكل التي واجهتها شبكات النوع الأول، ولكن هذا الانتقال لم يكن سهلاً، فعند محاولة الانتقال إلى البنية ذات المتحكمات المتعددة برزت مجموعة من المشاكل والتحديات من أهمها: عدد المتحكمات اللازمة لإعطاء أحسن أداء في الشبكة، وأماكن توضع هذه المتحكمات، بالإضافة إلى ضرورة وجود آلية فعالة لتوزيع الحمل بين هذه المتحكمات، أي طريقة تخصيص المهام الواردة إلى الموارد الموجودة بالشكل الذي يضمن أكبر فعالية في آلية عمل الشبكة، لذا حاولنا التطرق في هذا البحث إلى خوارزميات موازنة الحمل المستخدمة حالياً في بيئات SDN ذات

المتحكمات المتعددة، مع محاولة دراسة إيجابيات وسلبيات كل من الخوارزميات السابقة، والاستفادة من هذه الدراسة لاقتراح خوارزمية موازنة حمل جديدة، ومقارنتها مع الخوارزميات المستخدمة سابقاً، بهدف تحسين أداء الشبكة من ناحية ثلاثة عوامل رئيسية تمت دراستها في هذا البحث هي: إنتاجية الشبكة، معدل استخدام الموارد، واحتمال فشل المتحكم.

3- طرائق البحث ومواده:

يتطرق هذا البحث إلى الحاجة إلى الانتقال من الشبكات التقليدية إلى شبكات SDN، وبنية شبكات SDN، وعدم مواكبة البنية التقليدية المركزية لشبكات SDN للتطور الحاصل في عالم الشبكات، والحاجة إلى الانتقال إلى شبكات SDN بمتحكمات متعددة، مع ذكر الفوائد التي تحققها هذه البنية الموزعة، بالإضافة إلى أهم التحديات التي تتعرض لها، والنماذج المختلفة لشبكات SDN ذات المتحكمات المتعددة، كما تم التطرق إلى تعريف آلية موازنة الحمل، والعوامل التي تؤثر على هذه الآلية، وذكر أهم الخوارزميات المستخدمة في موازنة الحمل في الوقت الراهن، وأخيراً تم اقتراح آلية موازنة حمل جديدة ومقارنتها مع الخوارزميات المستخدمة سابقاً، وتم التطرق إلى التجارب التي أجريناها، والنتائج التي حصلنا عليها، وتوضيح التحسين الذي قدمته هذه الخوارزمية المقترحة على أداء الشبكة .

4- الحاجة إلى الشبكات المعرفة برمجياً:

برزت الحاجة في السنوات الأخيرة إلى مفهوم جديد للشبكات التقليدية بهدف معالجة المشاكل التي تعاني منها، كان الهدف الأساسي لشبكات SDN هو تبسيط عملية إدارة الشبكة وإضافة المزيد من المرونة إليها بالمقارنة مع الشبكات التقليدية. وقد تمت ملاحظة أربع مشاكل أساسية تعاني منها الشبكات التقليدية يمكن تلخيصها في:

1. **التعقيد وإدخال الإعدادات الشبكية بشكل يدوي:** تعتبر عملية إدخال الإعدادات الشبكية هي عملية معقدة في الشبكات التقليدية، حيث إن كل جهاز شبكي يتم إدخال الإعدادات إليه بشكل يدوي وبالطريقة التي تناسب الشركة المصنعة لهذا الجهاز، وقد تختلف هذه العملية بين جهاز وآخر على نفس الشبكة تبعاً للشركة المصنعة له،

بالإضافة إلى أن بنية الشبكة تتغير بشكل متكرر، فمن الممكن أن تتعرض إحدى الوصلات للفشل، أو أن نضطر إلى إضافة أو إزالة بعض الأجهزة الشبكية، كل هذه الحالات تضطرنا إلى إدخال الكثير من الإعدادات الشبكية بشكل يدوي وهو ما يعتبر عملية معقدة ستقودنا إلى حدوث مزيد من الأخطاء في إعدادات الشبكة [1].

2. الحالة الديناميكية للشبكة: يزداد حجم الشبكات بشكل مطرد يوماً بعد يوم، مما يزيد من تعقيد عمليات إدارة الشبكات، علاوةً على ذلك فإن ظهور تقنيات حوسبة جديدة مثل الحوسبة السحابية والضبابية وغيرها، بالإضافة إلى تقنيات الشبكات الافتراضية جعل الشبكات تصبح أكثر ديناميكية لأن المضيفين يتحركون باستمرار ويتصلون ويغادرون بشكل متكرر، هذا جعل حركة المرور وظروف الشبكة تتغير بشكل أسرع وبطريقة ملحوظة [2].

3. اختلاف مستويات التعقيد: تتطلب الشبكات الكبيرة في يومنا الحالي القيام بالعديد من المهام ذات مستوى تعقيد عالٍ، وهذه المهام يتم تنفيذها في الشبكات التقليدية عن طريق واجهات ذات مستوى تعقيد منخفض وميزات أقل، وهذا بسبب الترابط الوثيق بين طبقة الإدارة وطبقة البيانات، حيث تكون العديد من مزايا الإدارة متضمنة في عتاديات الأجهزة الشبكية.

4. عدم التجانس في أجهزة الشبكة: تتكون الشبكات الحالية من عدد كبير من الأجهزة غير المتجانسة مثل الموجهات والمبدلات وغيرها، ويملك كل جهاز من هذه الأجهزة تكوينها الخاص وإعداداتها الخاصة وكل منها يعمل وفقاً لبروتوكولات محددة قد تختلف من جهاز إلى آخر، مما يزيد من تعقيد إدارة الشبكة ويظهر عدم كفاءة هذه العملية.

أوضحت هذه الأسباب جميعها أن إدارة الشبكة عملية تزداد تعقيداً يوماً بعد آخر، وخاصةً في الشبكات التقليدية التي لا تتسم بالمرونة الكافية للتعامل مع الحاجات المتزايدة للشبكات في الوقت الحالي والتي يجب أن تتصف بالديناميكية العالية. هذا الأمر دفع بمنظمة ONF لاقتراح نموذج جديد هو نموذج الشبكات القابلة للبرمجة SDN [3].

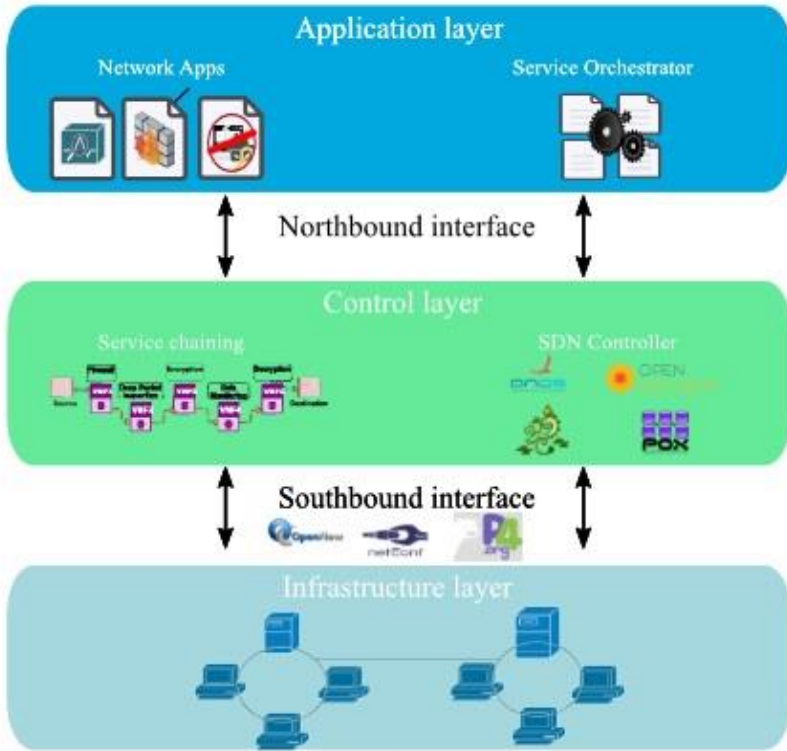
5- بنية الشبكات المعرفة برمجياً:

تعتمد شبكات SDN بشكل رئيسي على فصل طبقة التحكم عن طبقة البيانات، أي بشكل أساسي نقل عمليات اتخاذ القرار من العناصر الشبكية والتي هي عتاديات وتجميعها في عنصر مركزي وحيد هو المتحكم المبني على أساس برمجي، يتم اتخاذ كافة القرارات في المتحكم المركزي مما يجعل عملية إدارة الشبكة أكثر سهولة.

تتألف بنية شبكات SDN من ثلاث طبقات رئيسية و3 مكونات فرعية:

- **طبقة التطبيقات:** تحوي العديد من البرامج المسؤولة عن عمليات إدارة الشبكة وتهدف هذه الطبقة إلى تطوير العديد من التطبيقات وذلك من خلال الاستفادة من المعلومات المتوافرة كحالة الشبكة والمعلومات والإحصائيات المتوافرة حولها.
- **طبقة التحكم:** هي الطبقة الأذكى والأكثر أهمية في شبكات SDN والتي تحوي المتحكم الوحيد أو مجموعة المتحكمات الموجودة في شبكات SDN وهي التي تقوم باتخاذ القرارات في الشبكة.
- **طبقة البيانات أو طبقة البنية التحتية:** هي التي تحوي عدة أنواع من الأجهزة الشبكية من موجّهات ومبدلات وغيرها وظيفتها الأساسية هي المساعدة في تمرير التدفقات وتنفيذ القرارات المتخذة في طبقة التحكم [4].
- ✓ **الجسر الشمالي:** هو صلة الاتصال بين طبقة التحكم وطبقة الإدارة والذي تكون عادة مجموعة من واجهات برمجة التطبيقات API.
- ✓ **الجسر الجنوبي:** هو صلة الاتصال بين طبقة التحكم وطبقة البيانات والذي يكون عادةً عبارة عن بروتوكول يتم به التخاطب وهو في أغلب الأحيان بروتوكول OpenFlow
- ✓ **الجسرين الشرقي والغربي:** هما صلة الاتصال التي يتم فيها التخاطب بين المتحكمات المتعددة في طبقة التحكم وعادة تكون عبارة عن بروتوكولات شبكية لتبادل الرسائل مثل Border Gateway Protocol (BGP) و Open Shortest Path First (OSPF) [5].

حيث يبين الشكل (1) بنية الشبكات المعرفة برمجياً:



الشكل (1) بنية الشبكات المعرفة برمجياً.

6- الحاجة إلى البنية الموزعة بمتحكمات متعددة:

تطبق هيكلية شبكات SDN مبدأ التحكم المركزي من الأعلى إلى الأسفل، في هذه الحالة فإن متحكم SDN مع تطبيقات SDN المثبتة تعمل معاً لتتحكم وتنظم عمل المبدلات في الشبكة. عند تطبيق التحكم المركزي فإن هذا التصميم يعطينا العديد من الفوائد مثل إمكانية إدارة الشبكة بكفاءة والتعامل مع التغيرات الديناميكية في بنية الشبكة بشكل سهل وسريع. هناك تحقيقان ممكنان للوصول إلى تصميم التحكم المركزي، أولهما من خلال متحكم وحيد في بنية مركزية، والثاني من خلال عدة متحكمات في بنية موزعة. عند وجود متحكم وحيد في بنية مركزية يوجد متحكم SDN وحيد في الشبكة ككل، يقوم هذا المتحكم بجميع عمليات اتخاذ القرارات في الشبكة ويتحكم بطريقة مركزية بجميع المبدلات

في الشبكة بما أنها جميعها متصلة به. العديد من تطبيقات SDN سابقاً استخدمت هذا النموذج ولكنه للأسف عانى من مشكلتين أساسيتين لا يمكن إغفالهما وهما:

❖ **قابلية التوسع:** تعرف قابلية التوسع بأنها قدرة الشبكة على النمو بشكل يليبي الازدياد المطرد في أعداد المستخدمين وطلباتهم، وتعرف قابلية التوسع في شبكات SDN بأنها قدرة المتحكم على معالجة الطلبات الواردة من المبدلات المتصلة فيه. تملك متحكمات SDN موارد محدودة عند معالجة أعداد كبيرة من الطلبات مما يمنع من معالجة جميع الطلبات الواردة. حاول الباحثون حل هذه المشكلة من خلال تقييد عدد الطلبات الواردة إلى المتحكم، ولكن هذه الاستراتيجية لم تعمل بشكل جيد وخاصةً أنها تعتمد على منع خروج الطلبات من المبدلات، مما يعني أن جزءاً من عملية اتخاذ القرار عاد إلى المبدلات، وهذا الأمر كان ضد الفكرة الأساسية من شبكات SDN والتي تقوم على نقل جميع عمليات اتخاذ القرارات إلى المتحكم، لذا كان لا بد من وجود حل آخر وقد كان الحل الوحيد هو إضافة عدد من المتحكمات إلى الشبكة مما دفعنا بالحاجة للانتقال إلى البنية الموزعة بمتحكمات متعددة.

❖ **نقطة الفشل الوحيدة:** عند وجود متحكم SDN مركزي وحيد سوف يعاني من نقطة الفشل الوحيدة، وفي حال فشل المتحكم المركزي فإن جميع المبدلات سوف تفقد القدرة على تمرير الحزم الجديدة الواردة، وفي النهاية سوف تتعطل الشبكة بشكل كامل. وقد تم اقتراح حلين، كان الحل الأول يعتمد على أنه في حال حصول نقطة الفشل الوحيدة يعود اتخاذ القرارات إلى المبدلات، وتنتقل الشبكة من شبكة SDN لتعود إلى شبكة تقليدية، ولكن هذا الحل كان مكلفاً لأنه في حال العودة إلى الشبكة التقليدية فإن المشاكل المتعلقة بمثل هذا النوع من الشبكات، والتي ذكرناها سابقاً سوف تعود إلى الواجهة. بينما يتمثل الحل الثاني باعتماد شبكات SDN ذات بنية موزعة مع متحكمات متعددة وقد كان هذا الحل هو الحل الأفضل [6].

7- فوائد البنية الموزعة:

في شبكات SDN الكبيرة يمكن أن نحتاج إلى تقسيم الشبكة إلى مجالات مختلفة أصغر، عندها يحتاج كل مجال من هذه المجالات إلى متحكم واحد على الأقل ليشرف عليه، في هذا السيناريو فإننا نحتاج إلى عدة متحكمات لإدارة الشبكة بشكل كامل. إن وجود عدة متحكمات في الشبكة يساهم في زيادة استقرار الشبكة حيث إن المبدلات من الممكن أن تستمر بالعمل حتى في حال فشل أحد المتحكمات أو إحدى الوصلات إلى هذه المتحكمات، يمكن تحقيق البنية الموزعة بحيث يكون التحكم مركزي بشكل منطقي ولكنه موزع بشكل فيزيائي [7]، وهناك مجموعة من الفوائد الإضافية تتلخص في:

أ. **سهولة الإدارة:** تعتبر الإدارة الفعالة عاملاً مهماً لاستخدام متحكمات متعددة، حيث إن قدرات المتحكم الوحيد تعتبر محدودة بالمقارنة مع وجود متحكمات متعددة، حيث إنه من الصعب على متحكم وحيد معالجة جميع الطلبات الواردة في الشبكات الكبيرة ذات المجالات المختلفة، بالنظر إلى الميزات المختلفة الموجودة في عدة متحكمات، فإن وجود عدد من المتحكمات يمكن أن يطور من كفاءة إدارة الشبكة وخاصة في الشبكات الكبيرة ذات المجالات المتعددة [8].

ب. **قابلية التوسع:** إن نهج OpenFlow الحالي سوف يؤدي إلى زيادة الحمل بشكل كبير في حال وجود متحكم وحيد، مما يقلل من قابلية التوسع في الشبكة، بينما البنية الموزعة في حال وجود متحكمات متعددة يمكن أن تكون حلاً مثالياً، حيث يمكن إزالة وإضافة المتحكمات من وإلى الشبكة بسهولة كبيرة وبشكل ديناميكي، بالإضافة إلى أنه يمكن أن نقوم بتوزيع الحمل بين المتحكمات المختلفة لتجنب مشكلة عنق الزجاجة الموجودة في المتحكم المركزي الوحيد. كما أن الهيكلية الموزعة يمكن أن تكون تصميمًا مناسباً لدعم عدد أكبر من المستخدمين.

ج. **تقليل زمن الاستجابة:** هناك العديد من التطبيقات الحساسة لزمن الاستجابة، فإذا تمت معالجة جميع الطلبات من قبل متحكم وحيد، هذا سوف يؤدي إلى زيادة الحمل

على هذا المتحكم وسوف تتم معالجة الطلبات بشكل متسلسل الواحد تلو الآخر، مما سيؤدي إلى زيادة زمن استجابة هذه الطلبات بشكل كبير، وعند وجود متحكمات متعددة يمكن توزيع الحمل بينها أو تمرير الطلبات إلى المتحكم الأكثر ملاءمة مما يقلل من زمن الاستجابة للطلبات بشكل ملحوظ.

د . **المثانة:** إن بنية شبكات SDN بمتحكمات متعددة توفر العديد من نقاط الاستعادة والنسخ الاحتياطي من أجل تجنب مشكلة نقطة الفشل الوحيدة الموجودة في شبكات SDN ذات المتحكم الوحيد، وهذا سوف يزيد بالضرورة من مثانة الشبكة، عند فشل أحد المتحكمات يمكن للمتحكمات الباقية معالجة هذه المشكلة بتحمل العبء الناتج عن فشل هذا المتحكم من خلال توزيع حمله على باقي المتحكمات، حيث تتم معالجة هذه المشكلة بزيادة طفيفة على أحمال باقي المتحكمات، بالشكل الذي يسمح بالحفاظ على استقرار الشبكة. إن المتحكمات المتعددة يمكن أن تقلل بشكل كبير من احتمالية فشل الشبكة وخاصة في حال الأخطاء الناتجة عن فشل المتحكمات [9].

8- تحديات البنية الموزعة:

في بيئة SDN تعمل المتحكمات مع المبدلات جنباً إلى جنب لرفع مستوى أداء الشبكة، وخاصة الشبكات الكبيرة، ولرفع هذا المستوى كان من الأهمية بمكان تحديد تصميم مناسب لأماكن توضع المتحكمات المتعددة في الشبكة، والذي يعتبر من أهم القضايا الشائكة في شبكات SDN ذات المتحكمات المتعددة إلى يومنا هذا، بالإضافة إلى أماكن توضع المتحكمات يوجد قضايا أخرى مثل: المرونة وقابلية التوسع والأمن والثبات وتقليل زمن الاستجابة، وتعتبر هذه القضايا هي الأخرى قضايا مهمة لرفع مستوى أداء الشبكة [10].

يمكن تلخيص أهم التحديات بالنقاط التالية:

أ . الإدارة المركزية: تم تطوير المتحكمات المتعددة في شبكات SDN الكبيرة لموازنة الحمل في الشبكات، وعند زيادة عدد المتحكمات في الشبكة سوف يختلف مفهوم الإدارة المركزية، يجب استخدام متحكمات متعددة لتحقيق جودة خدمة عالية ، وذلك لأن سعة المتحكمات المتعددة تكون أكبر من سعة متحكم واحد، ولكن بالمقابل فإن المتحكمات المتعددة تملك مزايا مختلفة وبالتالي تتطلب طرق متعددة للتعامل معها، الأمر الذي يعتبر تحدياً بالنسبة لمطوري الشبكات.

ب . قابلية التوسع: بالرغم من أن البنية الموزعة تسمح بإضافة المتحكمات وزيادة حجم الشبكة بسهولة، إلا أن تحقيق عملية موازنة الحمل بين هذه المتحكمات تعتبر تحدياً مهماً لتحقيق جودة خدمة عالية.

ج . الاختلاف بين المتحكمات المتعددة: بسبب وجود عدد من المتحكمات فإن المشكلة الرئيسية هي عملية المزامنة بين هذه المتحكمات، ولذلك فإن عملية تصميم المتحكمات ومزامنتها تعتبر عملية معقدة [11].

د . توضع المتحكمات: إن وجود متحكم واحد في الشبكة له العديد من الميزات كسهولة الإدارة والتحكم ومراقبة الشبكة بشكل كامل من قبل نقطة وحيدة هي المتحكم المركزي. ولكن في نفس الوقت توجد مشاكل بالمقابل تتعلق بالثبات وقابلية التوسع، وبالتالي كلما زاد حجم الشبكة كلما أصبح الأداء أسوأ. ولكن عند حل هذه المشكلة باعتماد بنية SDN موزعة بوجود متحكمات متعددة تبرز مشكلة رئيسية هي اختيار عدد المتحكمات المطلوب وضعها في الشبكة، بالإضافة إلى اختيار أماكن توضع هذه المتحكمات والمسافة بين كل منها، ويمكن اعتبار المشكلتين الرئيسيتين في شبكات SDN ذات المتحكمات المتعددة تتلخص في سؤالين: الأول هو كم عدد المتحكمات المطلوبة في الشبكة لتحقيق أفضل أداء؟ والثاني هو في أي موضع من

الشبكة يجب وضع هذه المتحكمات؟ هذه المشاكل تعتبر معقدة جداً ومن الصعب الإجابة عليها، وهي تأخذ الحيز الأكبر من وقت الباحثين في وقتنا الراهن [12].

هـ . بروتوكولات الاتصال المختلفة: إن وجود عدة متحكمات مطلوب في البنية الموزعة ويؤثر بشكل رئيسي على جودة الخدمة في شبكات SDN، لحل مشكلة الاتصال بين هذه المتحكمات تم تطوير واجهات تدعى بالواجهات الشرقية والغربية والتي تتضمن مجموعة من البروتوكولات المستخدمة في عملية التخابط بين المتحكمات، لذا فإن إيجاد لغة برمجة وبروتوكولات مشتركة بين المتحكمات يعتبر تحدياً إضافياً في شبكات SDN.

و . عملية الجدولة بين المتحكمات المختلفة: تتحسن جودة الخدمة بوجود متحكمات متعددة، ولكن التحدي هنا هو عملية جدولة الطلبات بين هذه المتحكمات بالشكل الذي يعطي أفضل أداء للشبكة، ودون وجود حمل زائد على أي من المتحكمات في الشبكة. وكلما كانت عملية اتخاذ قرار الجدولة أسرع زادت جودة الخدمة [13].

9- نماذج الشبكات المعرفة برمجياً ذات المتحكمات المتعددة:

من الناحية النظرية في حال وجود متحكم وحيد في شبكة SDN يمكن لهذا المتحكم أن يتحكم بجميع المبدلات الموجودة في الشبكة فجميعها متصلة به، كما أنه يملك رؤية عامة للشبكة ككل، لذا في حالة المتحكمات المتعددة وللحصول على تحكم مركزي منطقياً لا بد من وجود متحكم واحد على الأقل يملك رؤية عامة للشبكة ككل في كل مجال من مجالات الشبكة، لذا فإن تبادل معلومات الشبكة المحلية بين متحكمات SDN المختلفة يعتبر أمراً لا مفر منه. يوجد نموذجان مستخدمان لتبادل معلومات الشبكة بين متحكمات SDN، وكل نموذج يقوم بتبادل هذه المعلومات بطريقة مختلفة عن الآخر.

❖ النموذج الهرمي: في هذا النموذج يكون أحد أو بعض المتحكمات (ولكن بالضرورة ليس جميع المتحكمات) في المجال تملك رؤية عامة للشبكة ككل. ويعرف هذا

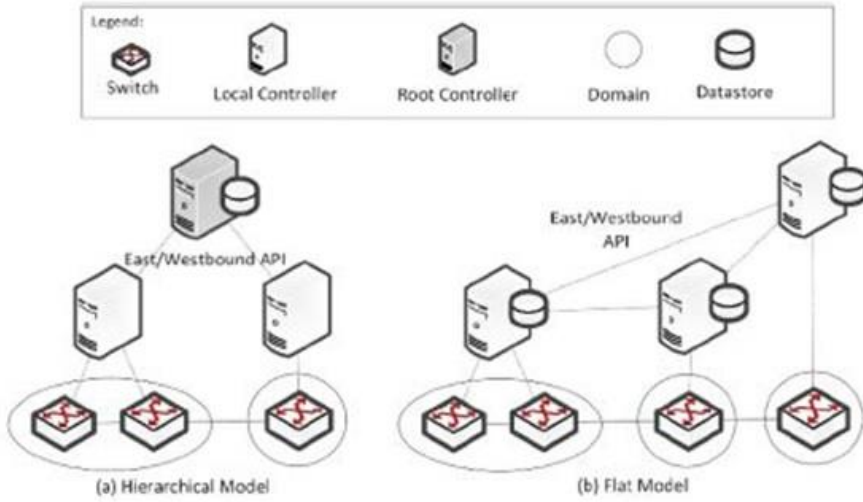
النموذج أيضاً بالنموذج العمودي لأن عمليات التحكم تتم من أعلى إلى أسفل عبر متحكمات SDN، يوضح الشكل (2) هذا النموذج، حيث يوجد متحكمي SDN محلين في كل مجال، بينما يوجد متحكم جذر هو المسؤول عن عمليات التنسيق بين المتحكمات المحلية. تكون معلومات الشبكة الكاملة مخزنة في المتحكم الجذر، لذا يجب على المتحكمات المحلية أن تطلب استعلام من المتحكم الجذر للحصول على المعلومات اللازمة قبل أي عملية اتخاذ قرار داخل مجالها المحلي، بسبب هذه الفكرة يمكن أن يدعى هذا النموذج بنموذج مخدم-زبون، حيث يعمل المتحكم الجذر كمخدم، وتعمل المتحكمات المحلية كزبائن. يمكن وضع أكثر من متحكم جذر وذلك لتجنب نقطة الفشل الوحيدة في المتحكم الجذر. لا تملك المتحكمات المحلية في هذا النموذج جسراً شرقياً وغربياً، أي أنها لا تملك وصلات مع باقي المتحكمات المحلية، ولكنها تملك وصلة واحدة فقط مع المتحكم الجذر الذي يقوم بعمليات التنسيق فيما بينها، ويحتفظ المتحكم الجذر بنسخة أو رؤية عامة عن بيانات الشبكة بشكل كامل، بينما تملك المتحكمات المحلية رؤية محلية للشبكة من منظورها الخاص فقط.

❖ **النموذج المسطح:** في هذا النموذج تملك جميع المتحكمات في المجال رؤية عامة عن حالة الشبكة، يعرف هذا النموذج أيضاً بالنموذج المسطح بما أنه لا يوجد معمارية هرمية بين متحكمات SDN، يوضح الشكل (2) هذا المبدأ حيث نلاحظ وجود ثلاث متحكمات SDN في المجال، هذه المتحكمات الثلاث تتحكم بالمبدلات، وتملك جميعها رؤية عامة عن حالة الشبكة، كما تملك هذه المتحكمات جسرين شرقي وغربي مما يسمح بعملية الاتصال فيما بينها بشكل مباشر. يجب على أي متحكم قبل البدء بعملية إنشاء حالة الشبكة من استقبال المعلومات عن حالة الشبكة من جميع المتحكمات الموجودة في المجال، وعند حصول أي تغيير في مجال أحد المتحكمات يجب أن تتم مشاركة هذا التغيير إلى جميع المتحكمات الباقية بالشكل

الذي يسمح لهذه المتحكمات بتحديث معلومات الشبكة لديهم وملاحظة هذه التغيرات، إذا تمت هذه العملية بشكل صحيح وبدون حدوث مشاكل هذا سوف يؤدي بالضرورة إلى أن جميع المتحكمات في الشبكة تملك نفس الرؤية العامة عن حالة الشبكة. بسبب هذه الفكرة يمكن أن يدعى هذا النموذج أيضاً بنموذج الند للند حيث يمكن لكل متحكم الوصول إلى المتحكمات الأخرى بشكل مباشر أثناء مشاركة معلومات الشبكة.

هناك طريقتان لتبادل معلومات الشبكة بين متحكمات SDN المختلفة في الشبكة، وهاتان الطريقتان هما:

- **الطلب الدوري:** في هذه الحالة يقوم كل متحكم بطلب المعلومات بشكل دوري من باقي المتحكمات في الشبكة، على سبيل المثال يقوم المتحكم بطلب معلومات جديدة عن حالة الشبكة من باقي المتحكمات كل 10 دقائق، سوف تتم عملية الطلب هذه بشكل دوري حتى لو لم يطرأ أي تغيير على الشبكة، هذا الأمر قد يؤدي إلى استلام نفس المعلومات أكثر من مرة مما يجعل هذه الطريقة غير فعالة بشكل كافي.
- **النشر والاشتراك:** حيث يقوم كل متحكم SDN بعملية نشر المعلومات الموجودة لديه إلى المشتركين، وعملية الاشتراك بالحصول على المعلومات من باقي المتحكمات في الشبكة، على سبيل المثال إذا كان أحد المتحكمات وليكن C1 يحتاج إلى معلومات من متحكم جار هو C2، عندها يجب على المتحكم C1 الاشتراك بنشرة المعلومات الصادرة عن C2، في هذه الحالة يعتبر C1 كمشارك و يعتبر C2 كناشر، وفي وقت لاحق سوف يقوم C2 بإعلام C1 عند حصول تغييرات في مجال C2، هذا يعني أن C2 يقوم بإعلام C1 فقط في حال حدوث تغييرات وليس بشكل دوري، مما يوضح أن هذه الطريقة أكثر فعالية من الطريقة الأولى [14].



الشكل (2) النموذج الهرمي و النموذج المسطح

10- موازنة الحمل:

تعرف موازنة الحمل بأنها العملية التي يتم توزيع مجموعة من المهام على مجموعة من الموارد وذلك بهدف جعل عملية معالجة هذه المهام أكثر كفاءة، حيث يمكن لهذه العملية أن تقلل من زمن الاستجابة للطلبات الواردة وتجنب الحالات التي تتعرض فيها بعض العقد في الشبكة إلى حمل زائد، بينما تكون هناك عقد أخرى في حالة سكون (Idle State).

10-1 آلية موازنة الحمل:

كما ذكرنا سابقاً يمكن تعريف موازنة الحمل بأنها العملية التي يتم توزيع وتخصيص الطلبات والمهام الواردة من الزبائن على الموارد المتاحة في الشبكة بفعالية وبالشكل الذي يضمن تخديم هذه الطلبات بأسرع طريقة ممكنة. تهدف عملية موازنة الحمل إلى تحسين نسبة استخدام الموارد لتجنب حدوث مشاكل الازدحام وفقدان البيانات. يمكن تطبيق عملية موازنة الحمل على الأجهزة العتادية، ويمكن تطبيقها أيضاً على البرمجيات. كما يجب على الجهاز الذي يقوم بموازنة الحمل أن يكون له بديل لتجنب نقطة الفشل الوحيدة. يوجد العديد من خوارزميات موازنة الحمل، قد يكون بعضها ستاتيكي والبعض

الآخر ديناميكي، النوع الأول يناسب البيئات التي تتسم بالثبات والاستقرار بما أنه يفقد المرونة وسرعة التكيف في حال حصول تغيرات ديناميكية في بنية الشبكة، في جميع الأحوال وبسبب أن سلوك المستخدمين لا يمكن التنبؤ به تتجه الخوارزميات الستاتيكية في توزيع الحمل بالفشل في أغلب الحالات. إن خوارزميات موازنة الحمل الديناميكية أكثر فعالية، لأن الطلبات يتم توزيعها على الموارد المتاحة تبعاً لحالة الشبكة في الوقت الراهن، وتبعاً للقواعد البرمجية المدخلة والتي تكون ديناميكية في هذه الحالة [15].

10-2- موازنة الحمل في شبكات SDN:

من ميزات شبكات SDN هي قدرتها على المحافظة على رؤية عامة للشبكة ككل، ولكن من جانب آخر فإن جميع عمليات اتخاذ القرارات تتم في المتحكم، مما سيسبب في زيادة الحمل على المتحكمات وزيادة تأخير زمن الاستجابة، لهذه الأسباب ظهرت الحاجة الماسة إلى تطبيق تكنولوجيا موازنة الحمل في شبكات SDN، مما سيسمح لعناصر الشبكة من تجنب مشاكل الازدحام ومعالجتها في حال حدوث حمل زائد على الشبكة. يجب على متحكم شبكة SDN تحسين معدل استخدام الموارد من خلال برمجيات معينة، هذا سيؤدي بالضرورة إلى زيادة الإتاحة في الشبكة وتحسن أدائها. إنتاجية الشبكة أيضاً سوف تتحسن عندما نقوم بتقليل زمن استجابة العناصر المشكلة للشبكة بالشكل الذي سوف يزيد من رضا الزبائن.

بروتوكول OpenFlow الذي يضمن عمليات إدارة الشبكة يعتبر العامل الأساسي في عمليات موازنة الحمل في شبكات SDN، حيث يعتبر هذا البروتوكول هو المسؤول عن تخزين جدول التدفقات، وكل مدخل في هذا الجدول يتألف من مجموعة من الحقول والعدادات، ويقابلها اتخاذ مجموعة من التعليمات والإجراءات الواجب تنفيذها في حال حدثت المطابقة مع أحد هذه المداخل في الجدول. باستخدام بروتوكول OpenFlow يمكن التحكم بالمداخل في جدول التدفق والحصول على الإحصائيات الخاصة بالشبكة وبكل مبدل وفقاً للقواعد والسياسات المطبقة في المتحكم الذي يستخدم هذا البروتوكول [16].

10-3- عوامل موازنة الحمل:

يمكن أخذ العديد من العوامل لتقييم نجاح آلية موازنة الحمل، ولكن أهم هذه العوامل التي أجمع عليها الباحثون يتلخص في أربع نقاط رئيسية هي:

1. **قابلية التوسع:** لا يمكن للآلية المركزية أن تتجح في الشبكات كبيرة الحجم والتي تنمو باستمرار لأنها تفتقر إلى المرونة، ولذلك كان لابد من استخدام بنية موزعة بوجود عدة مجالات وعدة متحكمات في الشبكة، وذلك من أجل الحفاظ على الإنتاجية في الشبكة. أحد أهم المعايير المستخدمة في التقييم هي إنتاجية الشبكة والتي يمكن حسابه بالقانون [17]:

$$F(N) = \emptyset(N) * T(N)/C(N) \quad (1)$$

حيث $F(N)$ تمثل إنتاجية الشبكة، $\emptyset(N)$ تمثل إنتاجية طبقة التحكم، $T(N)$ تمثل متوسط زمن الاستجابة للطلبات في الشبكة، $C(N)$ هي كلفة التطوير لطبقة التحكم.

2. **المرونة:** يقصد بهذا المعيار هو حالة الفشل التي يتعرض لها المتحكم، وهو من أهم المعايير في تقييم أداء آلية موازنة الحمل، حيث لا بد من ضمان عملية الانتقال إلى متحكم احتياطي في حالات فشل المتحكم الرئيسي، كما من الممكن أن تتأثر الشبكة في حال فشل أحد العقد أو الوصلات، يجب أن تضمن آلية موازنة الحمل تخصيص عادل للعقد على المتحكمات بالشكل الذي تقوم فيه هذه المتحكمات بمعالجة الأخطاء الناتجة عن الفشل في أحد الوصلات أو العقد بشكل سهل، وبدون أن تسبب زيادة في الحمل على أحد المتحكمات، كما يجب أن يتم أخذ المسافة والكلفة إلى المتحكم الاحتياطي في حال حدوث فشل في المتحكم الرئيسي بعين الاعتبار [18].

3. **معدل استخدام الموارد:** تم اقتراح عدة معايير من أجل قياس تأثير عملية موازنة الحمل على معدل استخدام الموارد ومن أهمها: معدل استخدام عرض الحزمة، حيث

يقوم هذا المقياس بقياس معدل النقل في الشبكة، ويعكس الحمل على الوصلات. يقوم المتحكم بحساب عدد البايتات المرسلة خلال فترتين متتاليتين وحساب المعدل في هاتين الفترتين، ثم يتم تقسيم هذا المعدل على العرض الأعظمي للحزمة ، وبالتالي نحصل على معدل استخدام عرض الحزمة، أو يمكن كفكرة عامة حساب معدل استخدام الموارد مثل الذاكرة وعرض الحزمة أو موارد أخرى مستخدمة في نقل المعلومات في الشبكة، من خلال حساب نسبة استخدام هذا النوع من الموارد خلال فترة معينة وتقسيمه على السعة الأعظمية لهذا المورد، وكلما زادت هذه النسبة زاد معدل استخدام المورد، وبالتالي فإن الشبكة تصبح تعمل بكفاءة أعلى [19].

4. جودة الخدمة: يوجد العديد من العوامل المهمة في حساب عامل جودة الخدمة، ولكن يمكن تلخيص أهم هذه العوامل في النقاط التالية:

✓ **زمن التأخير:** وهو الزمن الذي تحتاجه الحزمة للوصول من المصدر إلى الهدف عبر الشبكة، وهو يعتمد على الزمن الذي يستغرقه المبدل في تمرير الحزمة الواردة إليه بالإضافة إلى حالة الازدحام في الشبكة، وحجم البيانات المرسلة. لحساب زمن التأخير في مسار ما يجب حساب مجموع التأخيرات في كل وصلة من وصلات الموجودة على طول هذا المسار، ويعكس هذا الزمن حالة الشبكة، ويحدد إذا كانت مزدحمة ام لا، كما يوجد زمن آخر يدعى زمن الإنهاء وهو الزمن المستغرق في عملية اتخاذ القرارات في المتحكم بالشبكة، والذي قد يختلف تبعاً لخوارزمية موازنة الحمل المتبعة.

✓ **زمن الاستجابة:** يعرف بأنه الزمن المستغرق بين الطلب وحدث الاستجابة، عندما تكون الشبكة مزدحمة، أو عند وجود آلية ضعيفة لموازنة الحمل سيتجاوز معدل إنشاء الحزم السعة العظمى لموارد الشبكة، مما سيزيد من أرتال الانتظار وبالتالي سوف يزيد حكماً زمن الاستجابة. يوجد عامل مهم يؤثر على زمن

الاستجابة وهو درجة انشغال المتحكم والتي يمكن حسابها من خلال حساب الفرق بين زمن استجابة المتحكم في الوقت الراهن مع معدل استجابته العام. إن حساب زمن الاستجابة يعطي انطباعاً واضحاً عن مدى فعالية آلية موازنة الحمل المطبقة.

✓ **معدل فقدان الحزم:** يشير هذا العامل إلى عدد الحزم المفقودة خلال فترة الإرسال، ويعكس مقدار ازدحام المبدلات والوصلات في الشبكة، حيث إن المبدلات يمكن أن تكون مشغولة بشكل كبير وأن تعاني من حمل زائد، الأمر الذي يضطرها إلى تجاهل ورمي بعض الحزم الواردة. يمكن حساب عدد الحزم المفقودة في مسار ما من خلال طرح عدد الحزم المرسل من عدد الحزم المستقبل، كما يمكن حساب معدل فقدان الحزم لمسار ما من خلال تقسيم عدد الحزم المفقودة على عدد الحزم الكلي المستقبل في هذا المسار.

✓ **الخرج:** يعرف بأنه سرعة معالجة البيانات في عقد الشبكة، ويمكن حسابه من خلال حساب كمية أو حجم البيانات المرسل من المصدر إلى الهدف خلال واحدة مختارة من الزمن، وإن اختيار آلية موازنة حمل ناجحة سوف تؤدي إلى زيادة هذا العامل إلى أكبر رقم ممكن [20].

10-4- خوارزميات موازنة الحمل:

تلعب خوارزمية موازنة الحمل المستخدمة دوراً رئيسياً في أداء الشبكة، حيث إن اختيار خوارزمية موازنة حمل مناسبة وفعالة سوف يؤدي بالضرورة إلى تحسين أداء الشبكة، من خلال تقليل زمن الاستجابة للطلبات الواردة، وزيادة عدد الطلبات التي تتم معالجتها في الشبكة دون فشل، وتخفيض نسبة احتمال فشل أحد المتحكمات نتيجةً لتعرضه إلى حمل زائد، وبعض من أهم خوارزميات موازنة الحمل المستخدمة في وقتنا الحالي هي:

➤ **الخوارزمية العشوائية:** تقوم هذه الخوارزمية بتوزيع الحمل على المتحكمات بغض النظر عن حالة الشبكة، حيث تقوم بتوزيع الحمل بشكل عشوائي على الموارد المتوفرة بشكل عشوائي. تسند هذه الخوارزمية الطلبات القادمة إلى أحد الموارد وليكن i بشكل عشوائي و باحتمال $1/N$ ، حيث N هو عدد الموارد الموجودة في الشبكة، خلال هذه العملية لا يملك المتحكم المسؤول عن عملية موازنة الحمل أي معلومات عن حالة الشبكة والحمل الموجود عليها، وقد بينت الدراسات أن هذه الخوارزمية تعطي النتيجة الأسوأ من بين خوارزميات موازنة الحمل الأخرى وخاصة من ناحية زمن الاستجابة.

➤ **خوارزمية القادم أولاً يخدم أولاً:** تقوم هذه الخوارزمية على مبدأ أن الطلب الوارد أولاً يعطى إلى المورد الأكبر سعة، حيث يتم تخزين رتل للطلبات الواردة ريثما يتم إسنادها إلى مورد الشبكة في حال عدم توافر موارد متاحة. بالرغم من سرعة وبساطة خوارزمية موازنة الحمل هذه، إلا أنها أعطت نتائج سيئة بالنسبة لزمن الاستجابة، وذلك لأنها لا تأخذ بعين الاعتبار حجم المهمة الواردة ولا أولويتها [21].

➤ **الخوارزمية الدائرية:** تقوم هذه الخوارزمية بتوزيع الطلبات الواردة إلى الموارد المتاحة بشكل متساوٍ ووفقاً لخوارزمية جدولة دائرية. عند وصول أول طلب يقوم المتحكم المسؤول عن عملية موازنة الحمل بإسناد هذا الطلب إلى المورد الأول في قائمته، ويتم إسناد الطلب الثاني إلى المورد الثاني وهكذا حتى يصل إلى المورد N ، حيث N تمثل عدد الموارد المتاحة، في حال ورود طلب جديد $N+1$ يقوم بإسناده إلى المورد الأول، وتتكرر العملية إلى النهاية. تملك جميع الموارد في هذه الخوارزمية أولوية متساوية بغض النظر عن الحزم والطلبات الواردة،

وبغض النظر عن الزمن الذي تستغرقه الموارد في الاستجابة، هذا ما يفسر زمن الانتظار الكبير الذي تعاني منه المهمات الواردة.

➤ **الخوارزمية الدائرية مع أوزان:** تعتبر هذه الخوارزمية هي تطوير للخوارزمية الدائرية، حيث تختلف عنها بأن يوجد لكل مورد وزن في البداية، ويحدد هذا الوزن حجم الحمل الذي يجب تخصيصه لكل مورد، أي عدد المداخل التي سوف يملكها كل مورد في رتل الانتظار الدائري، أي أنه إذا كان وزن المورد الأول 2 ووزن المورد الثاني 3 ، فإنه سوف يتم تخصيص أول طلبين إلى المورد الأول، والطلبات الثلاث التالية إلى المورد الثاني وهكذا. تعتبر هذه الخوارزمية خوارزمية ستاتيكية، فهي لا تأخذ بعين الاعتبار تطور الأحمال في الشبكة أو حدوث أي تغييرات على بنيتها. لا تراعي هذه الخوارزمية عمليات البعد الجغرافي حيث من الممكن إسناد طلب وارد من مبدل ما إلى أبعد مورد عن هذا المبدل مما قد يسبب بمرور هذا الطلب بعدد كبير من العقد للوصول إلى الهدف وهذا يؤثر بشكل كبير على زمن الاستجابة.

➤ **خوارزمية الحمل الأقل:** تعتبر هذه الخوارزمية هي خوارزمية ديناميكية، حيث يقوم المتحكم المسؤول عن عملية موازنة الحمل بحساب عدد الطلبات في كل مورد، حيث يقوم بإسناد الطلب الوارد إلى المورد الأقل حملاً، أي المورد الذي يملك أقل عدد من الطلبات، قد يتغير الحمل في كل مورد عند تخصيص طلب جديد إليه، أو عند الانتهاء من تنفيذ طلب أو تجاوزه للمهلة المحددة. تعتبر هذه الخوارزمية هي خوارزمية ديناميكية، إذ أنها توزع الطلبات بناءً على أحمال الموارد في الزمن الحقيقي، ولكن من سلبياتها أنها لا تأخذ بعين الاعتبار الاختلاف في السعة بين الموارد، مما يؤثر على أداء الشبكة بشكل عام. تعتبر

هذه الخوارزمية مناسبة للشبكات المؤلفة من مجموعة موارد متساوية السعة
[22].

11-النتائج والمناقشة:

تم اقتراح خوارزمية جديدة لموازنة الحمل وقد اقترحنا تسميتها بخوارزمية الترحيل باستخدام العتبات الثلاث، وتمت مقارنة هذه الخوارزمية الجديدة مع خمس خوارزميات لموازنة الحمل والمستخدمة في الوقت الراهن، تمت عمليات مقارنة هذه الخوارزميات وفقاً لثلاثة عوامل وهي: إنتاجية الشبكة، معدل استخدام الموارد، واحتمال فشل المتحكم، وقد أثبتت التجارب إن الخوارزمية المقترحة تقدم تحسناً ملحوظاً على الخوارزميات السابقة من ناحية العوامل الثلاثة المدروسة.

11-1- السيناريو المقترح:

تمت مقارنة ست خوارزميات لموازنة الحمل وهي: الخوارزمية العشوائية، خوارزمية القدام أولاً يخدم أولاً، الخوارزمية الدائرية، الخوارزمية الدائرية مع أوزان، خوارزمية الحمل الأقل، والخوارزمية الأخيرة التي اقترحناها والتي اقترحنا تسميتها بخوارزمية الترحيل باستخدام العتبات الثلاث، وقد تم استخدام متحكم OpenDayLight المكتوب بلغة Java.

تمت عمليات المقارنة وفقاً لثلاثة عوامل وهي: إنتاجية الشبكة، معدل استخدام الموارد، واحتمال فشل المتحكم. الأداة التي قمنا باختبار التجارب عليها تمتلك معالج Intel® Core™ i9 [8-core@2.3](#) GHz 9th generation Processor و ذاكر GB 16 من نوع 2666MHz DDR4، و قرص صلب بسعة 1 TB SSD بنظام تشغيل MacOS مع أداة افتراضية على برنامج Virtual Box تم عليها تنصيب نظام تشغيل Linux 22.04 – 64 bit.

تمت عمليات المقارنة باستخدام المحاكي الشبكي Mininet والذي تم تنصيبه على الأداة الافتراضية المذكورة سابقاً.

11-2- الخوارزمية المقترحة:

تم اقتراح الخوارزمية التي تمت تسميتها خوارزمية الترحيل باستخدام العتبات الثلاث، حيث تحدد هذه الخوارزمية ثلاث عتبات لتحديد حالة المتحكم، هذه العتبات هي: T_{min} , T_{med} , T_{max} حيث يتم إدخالها يدوياً وفقاً لسعة المتحكم، وتكون $T_{med} = T_{max} + T_{min} / 2$ ، باستخدام هذه العتبات الثلاث يصبح لدينا أربع حالات لحمل المتحكم أ وهي:

إذا كان حمل المتحكم i : $L(i) < T_{min}$ يكون المتحكم ذا حمل قليل.

أما إذا كان حمل المتحكم i : $T_{min} < L(i) < T_{med}$ يكون المتحكم ذا حمل عادي.

أما إذا كان حمل المتحكم i : $T_{med} < L(i) < T_{max}$ يكون المتحكم ذا حمل عالٍ.

أما إذا كان حمل المتحكم i : $T_{max} < L(i)$ يكون المتحكم ذا حمل مفرط.

يمكن حساب حمل المتحكم i من خلال العلاقة: $L(i) = L_c(i) + L_m(i)$

حيث: $L_c(i)$ تمثل مقدار استهلاك وحدة التحكم المركزية في المتحكم i ,

$L_m(i)$ تمثل مقدار استهلاك الذواكر في المتحكم i .

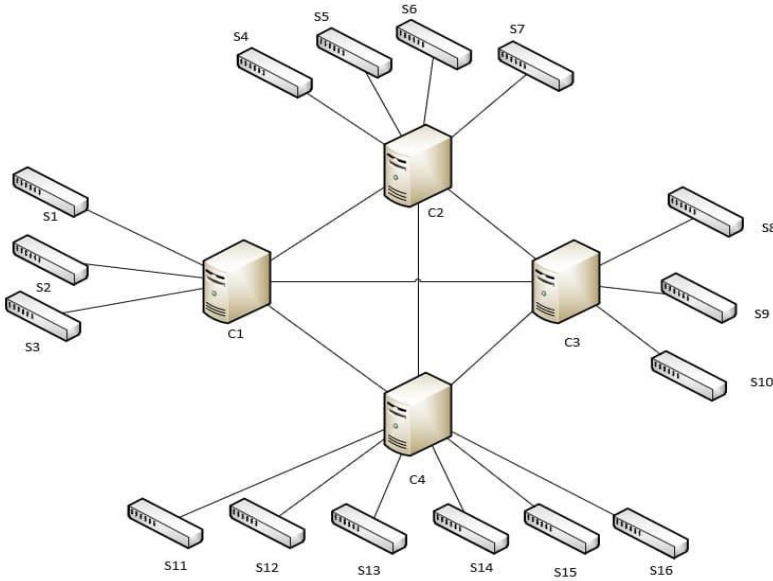
تتم عملية حساب حمل المتحكمات بشكل دوري، وإذا ظهرت حالة حمل مفرط أو عالٍ في أحد المتحكمات، يتم إطلاق عملية الترحيل، حيث يجب أن يتم ترحيل جزء من حمل المتحكم المفرط الحمل إلى المتحكم الأقل حملاً بين المتحكمات المتوفرة، عن طريق اختيار أحد المبدلات المتصلة بالمتحكم المفرط الحمل ونقل حمل هذا المبدل إلى المتحكم الذي تم اختياره، يتم اختيار المبدل الأقل حملاً في المتحكم المفرط الحمل، ليتم نقله إلى المتحكم المختار، وبعدها يتم تكرار العملية حتى نصل إلى حالة الاستقرار في الشبكة، بحيث تصبح جميع المتحكمات في حالة أحمال متقاربة. وفي حالة وجود أكثر من متحكم يعاني من نفس حالة الحمل المفرط أو العالي تكون الأولوية للمتحكم ذي الحمل الأعلى.

11-3- تحليل النتائج:

تم اقتراح بنية مؤلفة من 4 متحكمات هي المتحكمات C1,C2,C3,C4 متصلة مع 16 مبدل هي : S1,S2,...,S16.

في خوارزمية الترحيل باستخدام العتبات تم اقتراح حالة ابتدائية، حيث المتحكم C1 يتصل بالمبدلات S1,S2,S3 والمتحكم C2 يتصل بالمبدلات S4,S5,S6,S7 والمتحكم C3 يتصل بالمبدلات S8,S9,S10 بينما المتحكم الأخير C4 يتصل بالمبدلات S11,S12,S13,S14,S15,S16. تم توضيح البنية في الشكل (3):

باختيار عتبة $T_{min}=10\%$ و $T_{max}=90\%$ و بحساب T_{med} تنتج $T_{med}=50\%$.

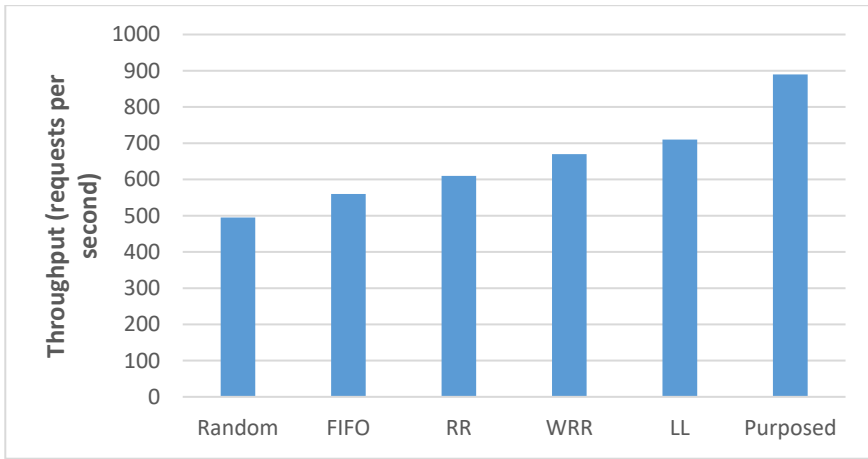


الشكل (3) البنية المقترحة.

11-3-1- السيناريو الأول: دراسة تأثير الخوارزميات المختلفة على إنتاجية الشبكة:

تعرف إنتاجية الشبكة بأنها عدد الطلبات التي تمت معالجتها خلال واحدة الزمن وقد اخترنا أن تكون ثانية واحدة، وكلما زادت إنتاجية الشبكة، كلما زاد عدد الطلبات التي تمت

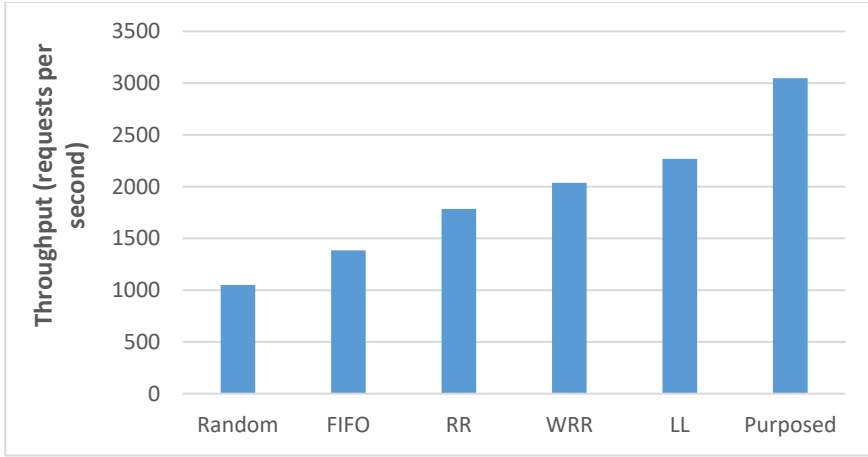
معالجتها في الشبكة دون فشل، وبالتالي تحسن أداء الشبكة بشكل عام، وإن استخدام خوارزمية موازنة حمل فعالة سوف يؤدي بالضرورة إلى رفع إنتاجية الشبكة. تمت دراسة كل من الخوارزميات الستة في أربع حالات هي حالة ورود 1000,4000,8000,16000 طلب في كل ثانية على التوالي وتم حساب إنتاجية الشبكة في كل من الحالات الأربعة كما بينت الأشكال (4,5,6,7). حيث يوضح الشكل (4) إنتاجية الشبكة في حالة ورود 1000 طلب كل ثانية:



الشكل (4) إنتاجية الشبكة في حالة ورود 1000 طلب كل ثانية.

بالنظر إلى الشكل (4) وهي حالة ورود 1000 طلب كل ثانية نجد أن الخوارزميتين العشوائية والقادم أولاً يخدم أولاً تعالج حوالي نصف الطلبات الواردة، أي أنها تعطي خرجاً يتراوح بين 50-60%، بينما تعالج الخوارزميات الدائرية، والدائرية بأوزان، وخوارزمية الحمل الأقل حوالي 60-70% من الطلبات الواردة، بينما الخوارزمية المقترحة تعالج حوالي 90% من الطلبات التي ترد إلى الشبكة.

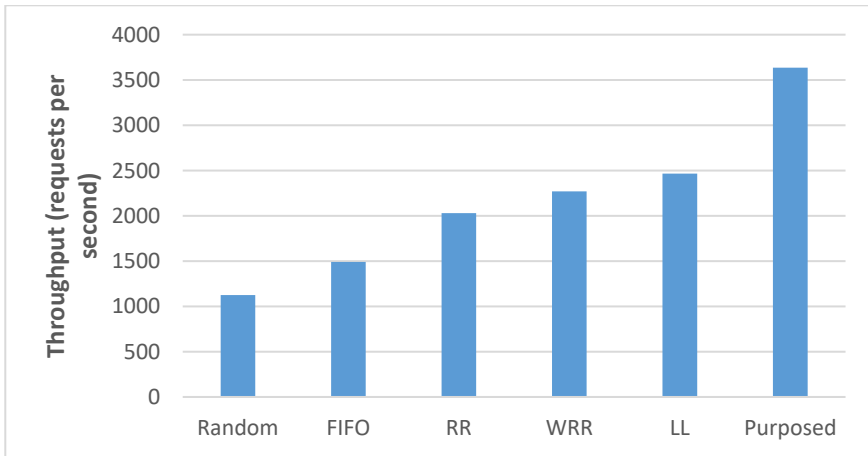
ومع زيادة عدد الطلبات الواردة خلال واحدة الزمن، والتي يوضحها الشكل (5) والذي يعبر عن إنتاجية الشبكة في حالة ورود 4000 طلب كل ثانية:



الشكل (5) إنتاجية الشبكة في حالة ورود 4000 طلب كل ثانية.

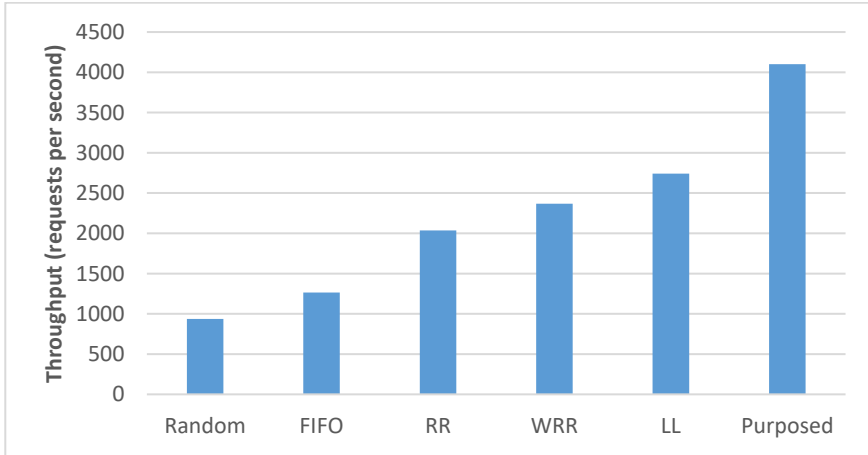
نستنتج من الشكل (5) وهي حالة ورود 4000 طلب كل ثانية نلاحظ انخفاض أداء الخوارزميات بشكل عام، حيث إن الخوارزمية العشوائية تعالج 25% من الطلبات الواردة، وخوارزمية لقدام أولاً يخدم أولاً تعالج حوالي 35% من الطلبات الواردة، بينما تعالج الخوارزميتان الدائرية والدائرية مع أوزان وخوارزمية الحمل الأقل حوالي 45% و 51% و 56% على التوالي، وتبقى الخوارزمية المقترحة هي الأفضل بنسبة معالجة 76% من الطلبات التي ترد إلى الشبكة.

و باستمرار زيادة الطلبات الواردة، تصبح إنتاجية الشبكة في حالة ورود 8000 طلب كل ثانية كما هو موضح في الشكل (6):



الشكل (6) إنتاجية الشبكة في حالة ورود 8000 طلب كل ثانية.

بالنظر إلى الشكل (6) وهي حالة ورود 8000 طلب كل ثانية، يستمر أداء الخوارزميات بالانخفاض، حيث إن الخوارزمية العشوائية تعالج 14% من الطلبات الواردة، وينخفض أداء خوارزمية القادم أولاً يخدم أولاً بشكل أكبر حيث تعالج حوالي 18% من الطلبات الواردة، بينما تعالج الخوارزميتان الدائرية والدائرية مع أوزان وخوارزمية الحمل الأقل حوالي 25% و 28% و 30% على التوالي، وتبقى الخوارزمية المقترحة هي الأفضل ولكن مع انخفاض بالإنتاجية، حيث تعالج بنسبة معالجة 45% من الطلبات الواردة. ومع الوصول إلى حالة 16000 طلب في الثانية تصبح إنتاجية الشبكة كما هي موضحة في الشكل (7):



الشكل (7) إنتاجية الشبكة في حالة ورود 16000 طلب كل ثانية.

من الشكل (7) نجد أن الخوارزميتين العشوائية والقادم أولاً يخدم أولاً تخدم عدداً أقل من الطلبات التي كانت تخدمها عند معدل ورود 8000 طلب في كل ثانية، وينخفض أداؤها حيث تعطي نسبة معالجة 6% و 8% على التوالي، وتحافظ الخوارزميتان الدائرية والدائرية مع أوزان على عدد الطلبات التي كانت تعالجها في حال ورود 8000 طلب، ولكن النسبة تتخفف بسبب أنها تعالج نفس العدد رغم زيادة المعدل من 8000 طلب إلى 16000 طلب كل ثانية، وهذا يعني انخفاض الإنتاجية إلى النصف، وتعطي نسبة

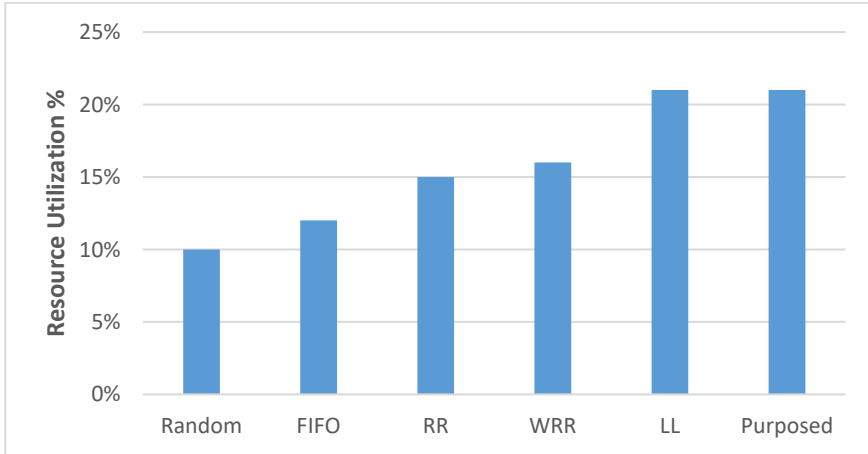
معالجة قدرها 12% و 14% على التوالي، كما تعطي خوارزمية الحمل الأقل 17% من الطلبات الواردة، بينما تعطي الخوارزمية المقترحة زيادة طفيفة في عدد الطلبات المعالجة، وتعطي نسبة معالجة قدرها 25%.

11-3-2- السيناريو الثاني: دراسة تأثير الخوارزميات المختلفة على معدل استخدام الموارد:

يعرف معدل استخدام الموارد بأنه نسبة انشغال المورد بالنسبة إلى سعته الكلية، وكلما زادت هذه النسبة كلما تحسن أداء الشبكة بشكل عام، لأنه يعني زيادة استغلال الموارد في معالجة الطلبات الواردة، وهذا الأمر يؤثر بشكل طردي على زيادة عدد الطلبات التي تمت معالجتها بنجاح، وبالتالي على أداء الشبكة.

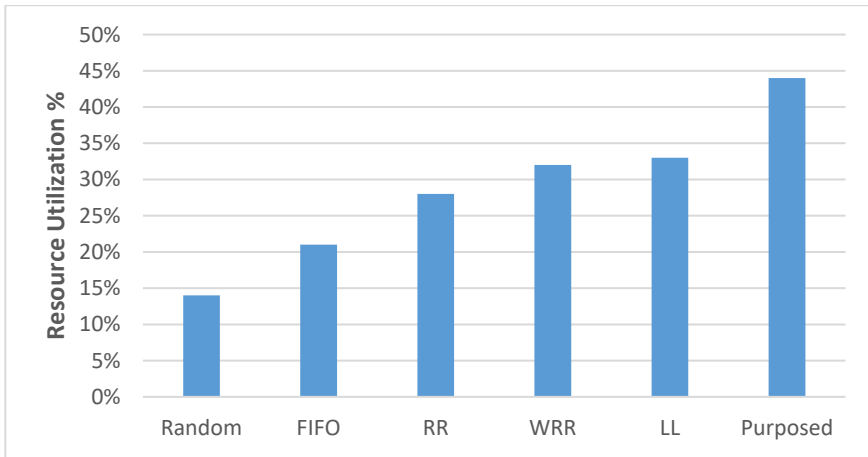
تمت دراسة كل من الخوارزميات الستة في أربع حالات هي حالة ورود 1000,4000,8000,16000 طلب في كل ثانية على التوالي وتم حساب معدل استخدام الموارد في كل من الحالات الأربعة بعد مرور 60 ثانية كما بينت الأشكال (8,9,10,11).

يوضح الشكل (8) معدل استخدام الموارد في حالة ورود 1000 طلب كل ثانية:



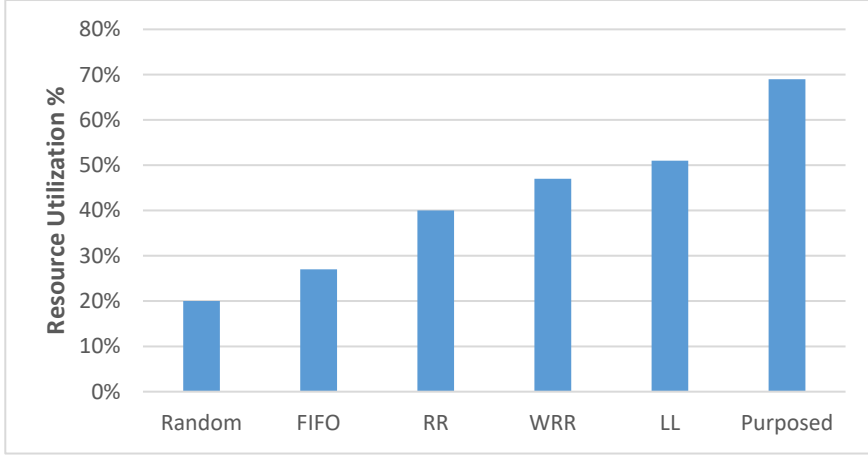
الشكل (8) معدل استخدام الموارد في حالة ورود 1000 طلب كل ثانية.

بالنظر إلى الشكل (8) وفي حال ورود 1000 طلب كل ثانية، نجد أن الخوارزميات العشوائية والقادم أولاً يخدم أولاً والدائرية والدائرية مع أوزان يكون معدل استخدام الموارد فيها متقارباً ويتراوح بين 10-15%، بينما خوارزميتي الحمل الأقل والخوارزمية المقترحة تعطي معدل استخدام متماثل بنسبة 21% من الموارد المتاحة، ونلاحظ انخفاض هذا المعدل بسبب أن عدد الطلبات وحجمها يعتبر قليلاً بالنسبة لسعة الموارد المتوفرة. ومع زيادة عدد الطلبات الواردة خلال وحدة الزمن والتي يوضحها الشكل (9) والذي يعبر عن معدل استخدام الموارد في حالة ورود 4000 طلب في الثانية الواحدة:



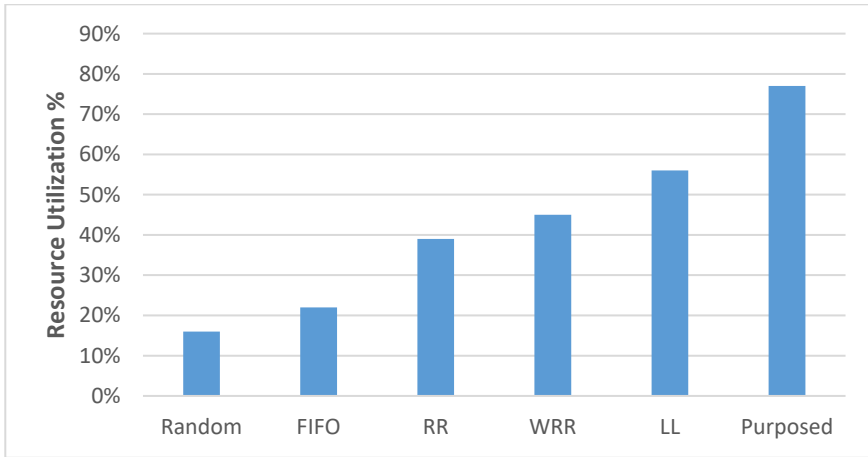
الشكل (9) معدل استخدام الموارد في حالة ورود 4000 طلب كل ثانية.

من الشكل (9) الذي يمثل حالة ورود 4000 طلب كل ثانية، نجد ارتفاع نسبة انشغال الموارد حيث تسجل الخوارزمية العشوائية، والقادم أولاً يخدم أولاً معدل استخدام قدره 14% و 21% على التوالي، بينما يتقارب أداء الخوارزمية الدائرية مع الخوارزمية الدائرية مع أوزان، وخوارزمية الحمل الأقل وتعطي نسب تتراوح بين 28-33% كمعدل استخدام، وتكون الخوارزمية المقترحة هي الأفضل مع معدل استخدام يبلغ 44%. وباستمرار زيادة الطلبات الواردة إلى 8000 طلب في الثانية، يصبح معدل استخدام الموارد كما هو موضح في الشكل (10):



الشكل (10) معدل استخدام الموارد في حالة ورود 8000 طلب كل ثانية.

بملاحظة الشكل (10) عند ورود 8000 طلب كل ثانية، نلاحظ ازدياد المعدل بشكل عام في جميع الخوارزميات المدروسة، حيث يبلغ 20% في الخوارزمية العشوائية، و27% في خوارزمية القادم أولاً يخدم أولاً، و40% في الخوارزمية الدائرية، كما تتشابه خوارزمتا الدائرية مع أوزان، والحمل الأقل في النتائج، حيث يعطيان معدل استخدام 47% للأولى و51% للثانية، ويكون أداء الخوارزمية المقترحة أفضل من الخوارزميات الباقية من حيث هذا العامل بمعدل 69%. ومع الوصول إلى حالة 16000 طلب في الثانية يصبح معدل استخدام الموارد كما هو موضح في الشكل (11):



الشكل (11) معدل استخدام الموارد في حالة ورود 16000 طلب كل ثانية.

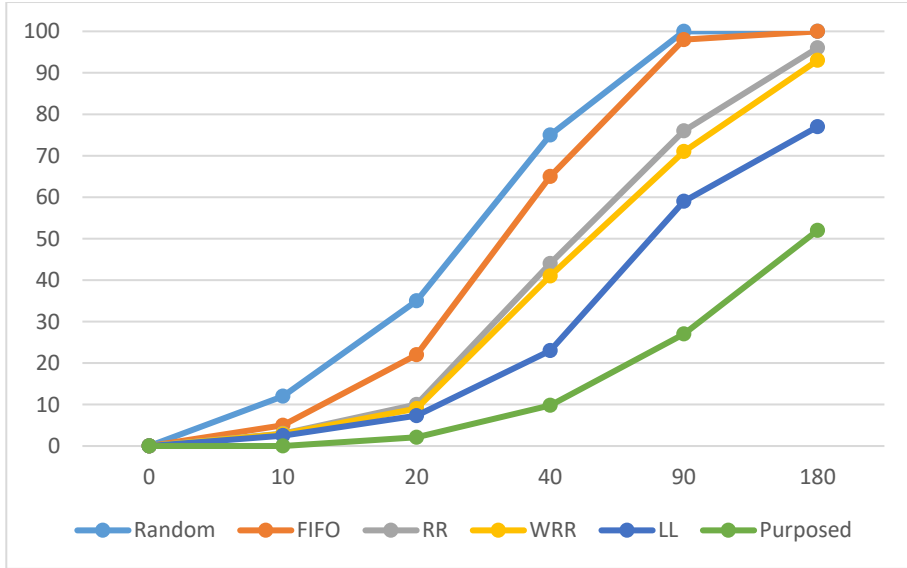
بقراءة النتائج من الشكل (11) في حال ورود 16000 طلب كل ثانية، نلاحظ انخفاض في معدل استخدام الموارد للخوارزميتين العشوائية والقادم أولاً يخدم أولاً، رغم زيادة عدد الطلبات الواردة، ويعود ذلك إلى حالة الفشل التي قد تصيب بعض المتحكمات وخروجها عن الخدمة، وتعطيان معدلات قدرها 16% و 22% للخوارزميتين السابقتين، كما نلاحظ أن الخوارزميتين الدائرية، والدائرية مع أوزان تحافظان على نفس المعدل تقريباً وتعطيان نتائج 39% و 45% على التوالي، بينما تظهر خوارزمية الحمل الأقل تحسناً طفيفاً بمعدل استخدام يبلغ 56%، و تظهر الخوارزمية المقترحة تحسناً أفضل وتعطي معدلاً قدره 77% من الموارد المتاحة.

11-3-3- السيناريو الثالث: دراسة تأثير الخوارزميات المختلفة على احتمال فشل

المتحكم:

يعرف هذا العامل بأنه النسبة المئوية المتوقعة لاحتمال فشل أحد المتحكمات، وكلما زادت هذه النسبة زادت احتمالات فشل المتحكمات وبالتالي زادت احتمالات انهيار الشبكة ككل، ومن الضروري مراعاة جعل أحمال المتحكمات متفاربة قدر الإمكان، وذلك لتجنب حالات فشل أحد المتحكمات بسبب الحمل الزائد، في حين وجود متحكمات أخرى بأحمال منخفضة.

تمت دراسة كل من الخوارزميات الستة في حال ورود 2000 طلب في كل ثانية، وتمت عملية الدراسة حتى 180 ثانية، وتم حساب احتمال فشل المتحكم تجريبياً، حيث تم إعادة التجربة 3 مرات لكل خوارزمية، وكانت النتائج مبينة في الشكل (12):



الشكل (12) احتمال فشل المتحكمات في الخوارزميات المختلفة.

من الشكل (12) نجد أن الخوارزميتين العشوائيتين، والقادم أولاً يخدم أولاً تعطيان أسوأ نتائج حيث إنه بعد مرور 20 ثانية يعطيان احتمالات فشل تتراوح بين 20-35%، وبعد مرور 40 ثانية ترتفع احتمالات الفشل إلى حوالي 60-70%، وبعد مرور 90 ثانية تصبح نسبة الفشل حوالي 100%، وذلك لأن كلتا الخوارزميتين لا تراعيان الأحمال الحالية للمتحكمات عند إسناد طلبات إليها، بينما تعطي الخوارزميتان الدائرية، والدائرية مع أوزان نتائج متقاربة، مع أفضلية طفيفة للخوارزمية الدائرية مع أوزان، وباحتمالات فشل تتراوح بين 40-44% في الثانية 40، وبين 70-75% في الثانية 90، ولكن كليهما تبلغ احتمالات الفشل فوق 90% عند الوصول إلى الثانية 180، ونلاحظ من الشكل أيضاً أن خوارزمية الحمل الأقل تعطي نتائج أفضل من الخوارزميات المذكورة السابقة وذلك لأنها تراعي الأحمال الحالية للمتحكمات، وتعطي احتمالات فشل لا تتجاوز 8% في الثانية 20، ولا تتجاوز 25% في الثانية 40، ولكن احتمال الفشل يرتفع إلى حوالي 59% في الثانية 90، ويبلغ 77% بعد مرور 180 ثانية، وفي النهاية تبقى الخوارزمية المقترحة هي الأفضل حيث إن بعد مرور 90 ثانية يبقى احتمال الفشل حوالي 27%، ويبلغ حوالي 50% عند وصولنا للثانية 180، وذلك لأنها تعتمد على عملية توزيع الحمل بين

المتحكمات في حال وصول أحد هذه المتحكمات إلى حمل زائد على عكس الخوارزميات الخمس التي تسبقها.

الخاتمة:

قمنا في هذا البحث بتوضيح الحاجة للانتقال من الشبكات التقليدية إلى شبكات SDN، وأيضاً الحاجة إلى استبدال نموذج شبكات SDN بمتحكم مركزي إلى شبكات SDN بمتحكمات متعددة، والفوائد المرجوة من استخدام البنية ذات المتحكمات المتعددة، كما تعرضنا إلى التحديات التي تواجهها هذه البنية الجديدة وكان أهمها: أماكن توضع المتحكمات، وخوارزميات موازنة الحمل المستخدمة لتحسين أداء الشبكة بشكل عام. كما قمنا في هذا البحث بالاهتمام بخوارزميات موازنة الحمل، وذلك عن طريق اقتراح خوارزمية جديدة لموازنة الحمل في شبكات SDN ذات المتحكمات المتعددة، وقمنا بمقارنة هذه الخوارزمية مع مجموعة من الخوارزميات المستخدمة سابقاً. تمت عملية الدراسة من خلال ثلاثة عوامل هي: إنتاجية الشبكة، ومعدل استخدام الموارد، واحتمال فشل المتحكم، وقد أثبتت الخوارزمية المقترحة تفوقها على الخوارزميات المستخدمة سابقاً من ناحية العوامل المدروسة.

فمن ناحية إنتاجية الشبكة، عالجت الخوارزمية المقترحة حوالي 90% من الطلبات الواردة عند ورود 1000 طلب في الثانية، وأعطت تحسناً قدره 20% عن أفضل الخوارزميات الأخرى في حالتها و 1000 و 4000 طلب في الثانية الواحدة، كما أعطت تحسناً قدره 15% في حالة ورود 8000 طلب في الثانية، وتحسناً قدره 8% في حالة ورود 16000 طلب في الثانية، وذلك بالمقارنة مع أفضل الخوارزميات الأخرى المدروسة. أما بالنسبة لمعدل استخدام الموارد، وفي حالة ورود 1000 طلب في الثانية تعطي الخوارزمية المقترحة نتائج متقاربة مع خوارزمية الحمل الأقل، ولكنها أفضل من جميع الخوارزميات الباقية، أما في حالات ورود 4000 و 8000 و 16000 طلب في الثانية

الواحدة، فقد أعطت الخوارزمية المقترحة تحسناً ملحوظاً بنتائج تحسين تقدر بـ 11%، 18%، 21% على التوالي، مقارنةً مع أفضل الخوارزميات الأخرى. أما بالنسبة للعامل الأخير وهو احتمال فشل المتحكم، نجد أنه بعد مرور 180 ثانية فإن الخوارزمية المقترحة تعطي تحسناً قدره حوالي 27% عن خوارزمية الحمل الأقل، وتحسيناً يقترب من 50% بالنسبة للخوارزميات الباقية. من جهة أخرى لا نستطيع إغفال أهمية عامل توازن المتحكمات في تحسين أداء الشبكة، و يلعب دوراً متكاملاً مع خوارزمية موازنة الحمل المستخدمة، لذا كان لابد من التوجه لدراسة هذا العامل مستقبلاً، وهذا هو موضع بحثنا في الوقت الحالي بعد الانتهاء من دراسة خوارزميات موازنة الحمل.

المراجع

- [1] HYOJOON KIM, NICK FEAMSTER; *Improving Network Management with software Defined Networking*; IEEE Communications Magazine, February 2013.
- [2] M. ABU SHARKH, M. JAMMAL, A. SHAMI, and A. OUDA; *Resource Allocation in a Network Based Cloud Computing Environment: Design Challenges*; IEEE Communications Magazine, September 2013, DOI: 10.1109/MCOM.2013.6658651.
- [3] FETIA BANNOUR, SAMI SOUIHI, and ABDELHAMID MELLOUK; *Distributed SDN Control: Survey, Taxonomy and Challenges*; *IEEE Communications Surveys & Tutorials*, December 2017, DOI: 10.1109/COMST.2017.2782482.
- [4] WENJUAN LIAB, YU WANGA, ZHIPING JIN, KEPING YUD, JIN LIA, and YANG XIANGA; *Challenge-based Collaborative Intrusion Detection in Software Defined Networking: An Evaluation*; *Digital Communications and Networks*, 9 September 2020, <https://doi.org/10.1016/j.dcan.2020.09.003>.
- [5] JUAN CAMILO CORREA CHICA, JENNY CUATINDIOY IMBACHI, and JUAN FELIPE BOTERO VEGA; *Security in SDN: A comprehensive survey*; *Journal of Network and Computer Applications*, 19 March 2020, <https://doi.org/10.1016/j.jnca.2020.102595>.
- [6] ESMAEIL AMIRI, EMAD ALIZADEH, and KHALILOLLAH RAEISI; *An Efficient Hierarchical Distributed SDN Controller Model*; 5th Conference on Knowledge-Based Engineering and Innovation, September 2019.
- [7] STEFAN SCHMID, JUKKA SUOMELA; *Exploiting Locality in Distributed SDN Control*; *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, ACM, 2013, pp. 121–126.
- [8] T. FRATCZAK, M. BROADBENT, P. GEORGOPOULOS, and N. RACE; *Homevisor: Adapting home network environments*, in: *Software Defined Networks*, 2013 Second European Workshop on, IEEE, 2013, pp. 32–37.
- [9] YUAN ZHANGA, LIN CUIA,B, WEIWANGC, and YUXIANG ZHANGA; *A Survey on Software Defined Networking with Multiple Controllers*; *Journal of Network and Computer Applications*, December 7, 2017, <https://doi.org/10.1016/j.jnca.2017.11.015>.

- [10] RAHIM MASOUDI, ALI GHAFARI; *Software defined networks: A survey*; Journal of Network and Computer Applications 67 (2016) 1–25, 26 March 2016, <http://dx.doi.org/10.1016/j.jnca.2016.03.016>.
- [11] MURAT KARAKUSA, ARJAN DURRESIA; *A Survey: Control Plane Scalability Issues and Approaches in Software-Defined Networking (SDN)*; Computer Networks November 10, 2016.
- [12] ASHUTOSH KUMAR SINGH, SHASHANK SRIVASTAVA; *A survey and classification of controller placement problem in SDN*; International Journal of Network Management · March 2018, DOI: 10.1002/nem.2018
- [13] SURENDRA KUMAR KESHARI, VINEET KANSAL, and SUMIT KUMAR; *A Systematic Review of Quality of Services (QoS) in Software Defined Networking (SDN)*; Springer Science+Business Media, 24 September 2020, <https://doi.org/10.1007/s11277-020-07812-2>.
- [14] YUSTUS EKO OKTIAN, SANGGON LEE, HOONJAE LEE, and JUNHUY LAM; *Distributed SDN controller system: A survey on design choice*; Computer Networks 121 (2017) 100–111, 11 April 2017, <http://dx.doi.org/10.1016/j.comnet.2017.04.038>.
- [15] AMIT S HANAMAKKANAVA, VIDYA S.HANDUR; *Load Balancing in Distributed Systems: A survey*; International Journal of Emerging Technology in Computer Science & Electronics(IJETCSE) ISSN: 0976-1353 Volume 14 Issue 2 –APRIL 2015.
- [16] JACOB H. COX, JR.1, JOAQUIN CHUNG, SEAN DONOVAN, JARED IVEY, RUSSELL J. CLARK, GEORGE RILEY, and HENRY L. OWEN; *Advancing Software-Defined Networks: A Survey*; IEEE Access, October 12, 2017, DOI: 10.1109/ACCESS.2017.276229.
- [17] JIE HU, CHUANG LIN, XIANGYANG LI, JIWEI HUANG; *Scalability of control planes for Software defined networks: modeling and evaluation*; IEEE 22nd international symposium of quality of service (IWQoS), Hong Kong. 2014. p. 147–52.
<https://doi.org/10.1109/IWQoS.2014.6914314>
- [18] SOUKAINA BOUZGHIBA, HAMZA DAHMOUNI, ANOUAR RACHDI, AND JEAN-MARIE GARCIA; *Towards an Autonomic Approach for Software Defined Networks: An Overview*; <https://www.researchgate.net/publication/309693700>, November 2017, DOI: 10.1007/978-981-10-1627-1_12.
- [19] YANING ZHOU, YING WANG, JINKE YU, JUNHUA BA, and SHILEI ZHANG; *Load Balancing for Multiple Controllers in SDN Based on Switches Group*; APNOM2017. IEEE. 2017.

- [20] YING-DAR LIN, CHIH CHIANG WANG, YI-JEN LU, YUAN-CHENG LAI, and HSICHANG YANG; *Two-tier dynamic load balancing in SDN-enabled Wi-Fi networks*; Springer Science+Business Media New York 2017, 9 April 2017, DOI 10.1007/s11276-017-1504-3.
- [21] GUDU BEKAMA HAILE, JIANXUN ZHANG; *Dynamic Load Balancing Algorithm in SDN-based Data Center Networks*; International Journal of Engineering Research & Technology (IJERT); ISSN: 2278-0181, Vol. 10 Issue 03, March-2021.
- [22] VIVEK SRIVASTAVA, RAVI SHANKAR PANDEY; *Load balancing for software-defined network: a review*; *International Journal of Computers and Applications*, 27 April 2021, <https://doi.org/10.1080/1206212X.2021.1919835>.

