Journal of Engineering Sciences and Information Technology
Volume (3), Issue (4): 30 Dec 2019
P: 83 - 99

ISSN: 2522-3321

مجلة العلوم الهندسية وتكنولوجيا المعلومات
المجلد (3)، العدد (4): 30 ديسمبر 2019 م
ص: 83 - 99

Improve the performance of peer-to-peer networks within publish/ subscribe systems by using PeerSim simulator within Eclipse Environment

Wael Abdulkarim Habeeb

Abdulkarim Assalem

Faculty of Electrical engineering || AL-Baath University || Syria

Abstract: Publish/ subscribe (pub/ sub) is a popular communication paradigm in the design of large-scale distributed systems. We are witnessing an increasingly widespread use of pub/ sub for a wide array of applications in industry, academia, financial data dissemination, business process management and does not end in social networking sites which takes a large area of user interests and used network bandwidth.

Social network interactions have grown exponentially in recent years to the order of billions of notifications generated by millions of users every day. So, it has become very important to access in the field of publishing and subscription networks, especially peer-to-peer (P2P) networks in many ways like the publication speed for events And the percentage of loss in the incoming events of the participants.

Peer-to-peer systems can be very large and include millions of nodes, those nodes join and leave the network continuously, and these characteristics are difficult to handle. The evaluation of a new protocol in a real environment, particularly in the early stages, was considered impractical. Hence the need for a simulator to perform such a function to facilitate the simulation of researchers and this emulator is an open source simulator running within the Eclipse environment.

In this research we have adopted a new method of selecting nodes within the table of vicinity protocol. This method is concentrated in that the far node increases the probability of its inclusion in the table more than the adjacent node. and The proposed network that uses the Polder Cast protocol was modelled using PeerSim software for modelling deployment and subscription networks within the eclipse environment so that the event delivery service is a Peer-2-Peer network and the method used to register is subject-based (Topic-Based).

experimental results showed noticeable improvement in the publication speed for events by 51.11% compared to the original design of the protocol. And The percentage of event loss was reduced by 20%.

Keywords: Peer-to-peer networks, publish-subscribe، systems, Polder Cast protocol.

تحسين أداء شبكات الند للند ضمن أنظمة النشر والاشتراك باستخدام محاكي Peersim ضمن بيئة Eclipse

وائل عبد الكريم حبيب

عبد الكريم السالم

كلية الهندسة الإلكترونية || جامعة البعث || سوريا

الملخص: يعتبر نظام النشر والاشتراك نموذج التواصل الأكثر شيوعاً في الانظمة الموزعة على نطاق واسع، حيث نشهد في أيامنا هذه انتشاراً متزايداً لاستخدام هذه الشبكات في مجموعة واسعة من التطبيقات تبدأ في مجال الصناعة والمجالات الأكاديمية العلمية والطبية

ونشر البيانات وادارة العمليات التجارية ولا تنتهي في مجال مواقع التواصل الاجتماعي التي باتت تشغل حيزاً واسعاً من اهتمامات المستخدمين ومن عرض حزمة الشبكة المستخدم.

تلك التفاعلات تضاعفت بشكل كبير عن الأعوام الماضية وأصبح يتم توليد مليارات الإشعارات اليومية من قبل ملايين المستخدمين على شبكات التواصل الاجتماعي. مما زاد من أهمية البحث في مجال شبكات النشر والاشتراك ضمن شبكات الند للند من عدة نواحي وأهمها سرعة نشر الأحداث للمشتركين ونسبة الفقد في الأحداث الواصلة للمشتركين.

يمكن أن تكون أنظمة الند للند على نطاق واسع جدًا وتضم ملايين العقد، تلك العقد تنضم للشبكة وتغادرها بشكل مستمر، ومن الصعوبة التعامل مع هذه الخصائص. حيث إن تقييم بروتوكول جديد في بيئة حقيقية، لا سيما في المراحل الأولى، كان يعتبر أمراً غير عملي. ومن هنا ظهرت الحاجة لوجود محاكي يقوم بمثل هذه الوظيفة لتسهيل قيام الباحثين بعمليات المحاكاة وهذا المحاكي simulator مفتوح المصدر يعمل ضمن بيئة Eclipse.

اعتمدنا في هذا البحث على طريقة جديدة في اختيار العقد ضمن جدول بروتوكول vicinity وهذه الطريقة تتركز في زيادة احتمالية ضم العقدة الأبعد إلى الجدول الخاص بالناشر أكثر من العقدة القريبة. وقد تم نمذجة الشبكة المقترحة والتي تقوم باستخدام بروتوكول Polder Cast باستخدام برنامج PeerSim الخاص بنمذجة شبكات النشر والاشتراك ضمن بيئة eclipse البرمجية بحيث تكون خدمة تسليم الحدث عبارة عن شبكة ند للند (Peer-2-Peer) والطريقة المُستَخدَمة لتسجيل الاشتراك معتمدة على الموضوع (Topic-Based).

وقد لاحظنا من خلال النتائج التجريبية تحسناً في سرعة إيصال الأحداث للمشتركين بمقدار 51.11% مقارنة بالتصميم الأصلي للبروتوكول. كما أنه تم تخفيض نسبة فقد الأحداث للعقد بمقدار 20%.

الكلمات المفتاحية: شبكات الند للند، أنظمة النشر والاشتراك، بروتوكول Polder Cast.

1- المقدمة:

يقصد بشبكة الند للند أن كل عقدة في هذه الحالة يمكن أن تلعب دور الناشر أو المشترك أو الوسيط ضمن شبكة النشر والاشتراك التي تعتمد على مبدأ إيصال الأحداث التي تهتم بها العقد ضمن الشبكة، ويتكون نظام النشر/ الاشتراك من ثلاثة عناصر (1,2) وهي:

- الناشر (Publisher): وهو الذي يقوم بإنتاج المعلومات.
- المشترك (Subscriber): وهو الذي يقوم باستقبال المعلومات.
- خدمة تسليم الحدث (Event Notification Service): وهي مؤلفة من وسيط (broker) أو مجموعة من الوسطاء، وهي تقوم بتسليم الحدث من الناشر إلى المشترك.

يبين الشكل (1) نموذجاً بسيطاً لنظام النشر/ الاشتراك.

الشكل (1) بنية بسيطة لنظام نشر واشتراك(1)

يقدم نظام النشر/ الاشتراك للمشتركين إمكانية التصريح عن اهتماماتهم بأصناف معينة من الأحداث، وبالتالي فإن أي حدث يتم نشره من قبل أحد الناشرين سيتم تسليمه إلى المشتركين الذين أبدوا اهتمامهم بهِ. بعبارة أخرى، يقوم الناشرون بتسليم المعلومات (الأحداث) إلى خدمة تسليم الحدث التي تقوم بدورها بتسليمها إلى المُشتَركين المهتمين بتلك الأحداث.

النهج الأكثر شيوعاً في هذه الانظمة هي اعتبار أن هذه الاجراءات تتم باستخدام مجموعة من الوسطاء يتم تنظيمها ضمن طبقة فوقية overlay، تقوم هذه الوسطاء بتخزين طلبات الاشتراك وتوجيهها للمشتركين المهتمين.

1-1- مشكلة البحث:

على الرغم من تطور بروتوكولات الاشتراك والنشر ضمن شبكات الند للند بشكل كبير في الآونة الأخيرة، إلا أن هذه البروتوكولات تهتم فقط بتوزع العقد ضمن الشبكة حسب زمن انضمامها دون مراعاة توضعها الفيزيائي، وذلك بسبب آلية الثرثرة التي يتبعها نموذج vicinity ضمن البروتوكولات السابقة مما يؤدي إلى انخفاض فعالية الحركة وازدياد أزمنة التأخير.

1-2- هدف البحث:

يهدف البحث لتصميم نظام النشر والاشتراك يحقق مجموعة واسعة من الخصائص المرغوبة تتعلق بالتوسعية العالية (scalability) والتأخير المنخفض والحمل القليل في الاتصالات. والعديد من هذه الخصائص ترتبط مع بعضها بشكل عكسي غالباً مما يستدعي المقايضة فيما بينها عند تصميم النظام، وهذا يجعل تحقيق التوازن الصحيح بينها هو تحد أساسي في منهاج البحث عند تصميم أي نظام نشر واشتراك. أي أن الهدف الرئيسي يكمن في تصميم نظام نشر واشتراك يحقق نسبة عالية من جميع الميزات السابقة دون أن يؤثر ذلك على أداء النظام من ناحية الميزات الأخرى.

1-3- أهمية البحث:

بسبب الأهمية الكبيرة والمتزايدة لأنظمة النشر والاشتراك والتي باتت تدخل في جميع مجالات الحياة اليومية، فإن تطوير وتحسين أداء تلك الشبكات بات من التحديات الأساسية في الوقت الراهن. ويعتبر تحسين سرعة نشر الأحداث لوصولها للمشتركين بأقل وقت ممكن، وتحسين توزيع الحمل على عقد الشبكة من التحديات الرئيسية التي يواجهها مزودي الخدمة والباحثون في مجال تطوير أداء شبكات النشر والاشتراك.

1- 4- أهم خصائص ومميزات أنظمة النشر/ الاشتراك:

تنبع قوة هذه الأنظمة وأهميتها من كونها تؤمن خواص عدم الارتباط الكامل Full Decoupling فيما يتعلق بالمكان وبالزمن وبالتزامن (3,4,5) كالتالي:

1- عدم الارتباط المكاني Space Decoupling:

تتجلى هذه الخاصة بكون الناشر والمشترك ليسا بحاجة لكي يعرف بعضهما بعضاً، وبدلاً من ذلك، تتولى خدمة تسليم الحدث مهمة تسليم الأحداث من الناشر إلى المشتركين المهتمين.

2- عدم الارتباط الزمني Time Decoupling:

تتجلى هذه الخاصة بكون الناشر والمشترك ليسا بحاجة لكي يكونا في حالة الفعالية (Active) في نفس الوقت، وهذا يعني أنه من الممكن أن يقوم الناشر بنشر حدث ما حتى عندما لا يكون المشترك في حالة اتصال، وبشكل مشابه، يمكن للمشترك أن يستلم حدثاً ما بغض النظر عن حالة الناشر.

3- عدم الارتباط من ناحية التزامن Synchronization Decoupling:

تتجلى هذه الخاصة بكون عمل الناشر لا يتوقف على عمل المشترك والعكس بالعكس.

5-1- تطبيقات أنظمة النشر والاشتراك:

عند الاطلاع على التطبيقات التي تستخدم أنظمة النشر والاشتراك ندرك أهمية تلك الأنظمة كونها تعدت مجالات الترفيه والتواصل الاجتماعي لتدخل المجال الأكاديمي والاقتصادي والصحي..... ومن تلك التطبيقات:

1- مكتبة Spotify الموسيقية:

- تعتبر خدمة ناجحة ومشهورة (6) لتدفق الموسيقى على الأنترنت حسب الطلب، والتي تؤمن الوصول لأكثر من 25 مليون مسار أغاني للمستخدمين المشتركين فيها في أكثر من 55 دولة حول العالم.
- إن شهرة هذا الموقع لم تأت من تأمينه لعدد هائل للمسارات الموسيقية فقط بل من خلال تسهيله لمشاركة ومتابعة نشاطات موسيقية متعددة لمستخدميه في الوقت الحقيقي.
- يمكن للمستخدم متابعة المستخدمين الآخرين، واكتشافهم عن طريق حسابهم على الفيس بوك.
- يمكن للمستخدمين متابعة الموسيقيين المسجلين في الموقع الشكل (2).

الشكل (2) مكتبة Spotify الموسيقية (6)

2- الألعاب التفاعلية ذات العدد الهائل من اللاعبينMassive Multiplayer Online Games :

في هذا التطبيق (2) عندما يقوم أحد اللاعبين بأداء فعل معين يؤثر على مجرى اللعبة، يجب أن يحصل بقية اللاعبون على تحديثات تتعلق بالتغيرات التي نتجت عن ذلك الفعل.

يُستَخدَم نظام النشر/ الاشتراك بحيث يقوم كل لاعب بإبداء اهتمامه بمنطقة واحدة أو أكثر من اللعبة وبالتالي يستقبل كل رسائل التحديث التي تشمل التغيرات الناتجة عن أفعال تقوم بها كيانات موجودة ضمن تلك المنطقة (أو مجموعة المناطق). الشكل (3)

الشكل (3) الالعاب التفاعلية (2)

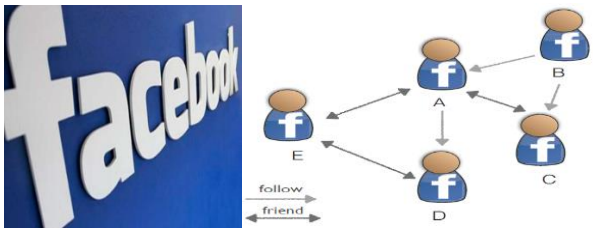
3- التلقيم المبسط(Really Simple Syndication) RSS:

- خدمة على مواقع الأنترنت تقوم بإيصال إي تغير نظامي في محتوى الويب إلى المشتركين باستخدام بريدهم الإلكتروني الشخصي(6).
- أغلب مواقع الأخبار والمدونات تقوم بإيصال أخبارها عن طريق الـ RSS لمن يريد من المشتركين. الشكل (4).

الشكل (4) خدمة RSS (6)

4- فيس بوك Facebook:

- يعتبر أهم مواقع التواصل الاجتماعي على الأنترنت (6) أهمية وانتشاراً.
- بلغ عدد المستخدمين المسجلين في فيسبوك 1.28 مليار مستخدم.
- حوالي 757 مليون مستخدم نشط يومياً. ويتم تناقل 4.75 مليار ملف وسائط متعددة كل يوم وهي أضخم شبكة تواصل اجتماعي على الأنترنت حتى يومنا هذا.
- تكون العلاقات في الفيس بوك ثنائية الاتجاه بشكل عام. أي على المستخدمين قبول صداقة كل منهم الآخر ليكونوا أصدقاء.
- بالإضافة لعلاقة الصداقة فإن فيس بوك يسمح بالعلاقة أحادية الاتجاه على غرار موقع تويتر. الشكل (5).



الشكل (5) مثال عن العلاقات بين المستخدمين في فيسبوك (6)

2- الدراسات المرجعية:

تم في هذا الصدد تصميم العديد من البروتوكولات العاملة في شبكات النشر والاشتراك بغية تحسين أداء تلك الشبكات وأهم هذه البروتوكولات:

2-1- بروتوكول clouds (1):

يعتمد هذا البروتوكول على تجميع العقد ذات الصفات المحددة ضمن مجموعات ويتكون من طبقتين:

1- طبقة غير منظمة unstructured layer: الهدف من هذه الطبقة تنظيم جميع العقد (التي تملك مصادر متوفرة) في النظام. وعلى الرغم من أن هذه الطبقة تنظم العقد إلا أن البنية الناتجة تبقى غير منظمة بالمعنى الحرفي لأنها لا تتبع أي جدول محدد يعتمد على معرفات العقد (ID'S)، وطريقة عمل هذه الطبقة تعتمد على خلق علاقات جيران بين العقد التي تملك نفس المصادر المتوفرة.

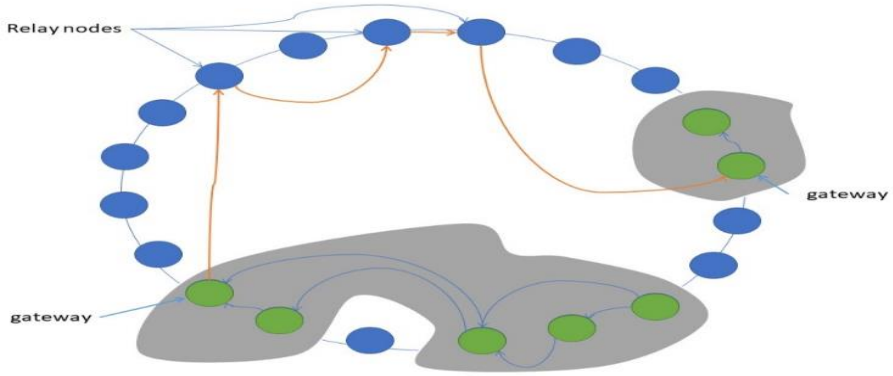
2- طبقة منظمة structured layer: تعمل هذه الطبقة بعد الطبقة السابقة أي بمجرد العثور على عقدة مجاورة بنفس صفات الموارد فإنها تباشر البحث عن عقد أخرى تملك نفس الصفات ضمن الطبقة غير المنظمة. أي الغرض من هذه الطبقة هو تسهيل الخطوة الأولى لإجراءات تحديد توضع الموارد.
لكن مشكلته الأساسية أنه لا يضمن وصول الاشتراكات إلى كل العقد.

2-2- بروتوكول Vitis (6,7):

يعمل هذا البروتوكول في البيئات الهجينة وهو بروتوكول قابل للتوسعية ويعمل على تقليل عدد الرسائل المكررة ولكنه يحوي على Overhead عند بعض العقد ويستخدم مسارات Relay كثيرة.

- يتم تنظيم العقد وفق مسار دائري كل عقده تعرف العقدة التي قبلها والتي بعدها وتضعها في جدول التوجيه، وعن طريق الثرثرة الدورية gossiping كل عقدة تعرف اهتمامات جيرانها، العقد القريبة من بعضها البعض والتي تتشارك نفس الاهتمام تقوم بتشكيل overlay، تحدد كل مجموعة عقدة للاتصال مع بقية المجموعات بمثابة بوابة عبور gateway.
- عندما يتم نشر حدث معين ينتقل عن طريق الثرثرة بين عقد المجموعة نفسها وينتقل إلى عقد المجموعات الأخرى المهتمة بنفس الموضوع عن طريق عقدة الـ gateway باستخدام عقد أخرى في الشبكة كوسطاء وتدعى هذه العقد عقد تسليم relay nodes.

والشكل (6) يوضح آلية عمل البروتوكول السابق:



الشكل (6) آلية عمل بروتوكول Vitis(6)

3-2- بروتوكول PolderCast (8):

هو بروتوكول تصميم شبكات اشتراك ونشر مستقل الطبقات. يتكون من ثلاث طبقات لكل منها جدول توجيه. حيث تحافظ هذه الطبقات بعد بناءها على بنية الشبكة.

1- تحوي الطبقة الأدنى بروتوكول الـ cyclon وتكون مسؤولة عن بناء الشبكة من خلال المحافظة على مجموعة من الروابط العشوائية للعقد.

2- أما الطبقة الأعلى يكون فيها الـ vicinity والذي يستخدم الأدنى منه cyclon لاكتشاف الاتصالات مع العقد العشوائية التي لديها مصالح مماثلة مع العقدة الحالية. حيث تعمل vicinity على إيجاد العقد المجاورة ذات المواضيع المشتركة عبر استخدام تابع تقارب، وبناء على الاحتمال فإن تابع التقارب لعقدة N من وجهة نظر عقدة P يتم قياسه بالاعتماد على مجموع الاحتمالات p لكل موضوع i يقومون بمشاركته، حيث Sn يشير إلى اشتراكات العقدة n، ويعطى تابع التقارب (3) هذا بالعلاقة:

$$proximity(N,P) = \sum_{i=0}^{n(S_N \cap S_P)} p_i \qquad (1)$$

3- ويحوي المستوى الأخير على الحلقة ring وهي المسؤولة عن الاشتراك والمنشورات.

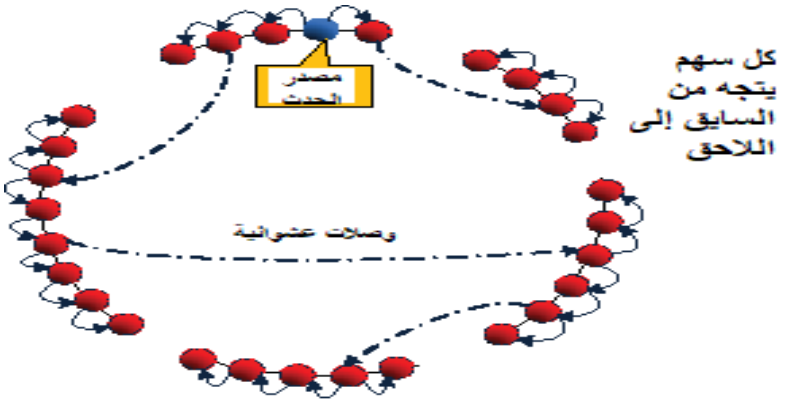
- تتواصل كل طبقة بشكل مستقل ومتزامن مع الطبقات الأخرى عن طريق عملية الثرثرة gossiping، والتي تقوم على اختيار الجار الأقدم حيث أن كل عقدة لها عمر ضمن تعريفها وهو يزيد مع كل دورة ثرثرة، ويصفر عند كل بداية اتصال جديد. وتتم رسائل الثرثرة gossip كل 10 ثواني. حيث تثرثر كل عقدة مع 10 جيران لها عن طريق الملف الخاص بها والذي يحوي تفاصيل العقدة ID واشتراكات العقدة.

لكي تقوم عقدة بالاشتراك بموضوع T تحاول الاتصال باي عقدة سابقة او لاحقة تجدها عن طريق الحلقة ring والذي يحوي أربع عقد أساسية لكل اشتراك...

- عقدتين يمثلان السابق واللاحق واللتان لهما ID أكبر وأصغر تماما من العقدة الحالية
- عقدتين لهما أكبر وأصغر ID من العقدة الحالية ويتشاركان بنفس الموضوع

يتم إيجاد هذه العقد الأربعة عن طريق عملية الثرثرة. حيث يحصل ring على العقد الأولية من vicinity الذي يقوم بإيجاد العقد المجاورة للعقدة والتي تهتم بنفس الموضوع باستخدام تابع التقارب وباستخدام أولوية كل موضوع بالنسبة للعقدة.

- يوضح الشكل (7) طريقة توزع العقد في بروتوكول Polder Cast.



الشكل (7) مخطط طريقة توزع العقد في بروتوكول PolderCast (8)

3- تصنيف أنظمة النشر/ الاشتراك:

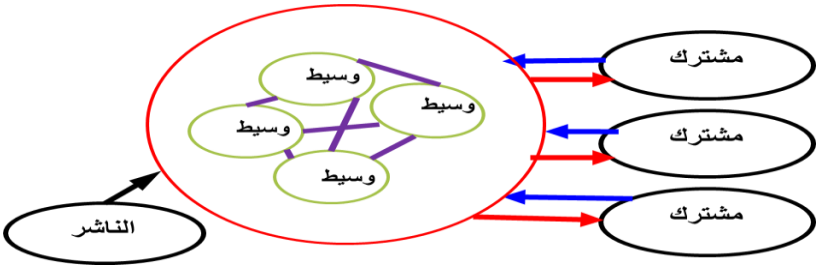
3-1 خدمة تسليم الحدث كوسيط مركزي:

يتصل الناشرون (8,9,10) والمشتركون بهذا الوسيط وعند نشر حدث ما يقوم الناشر بتسليمه إلى الوسيط المركزي الذي يقوم بدوره بتسليمه إلى المشتركين المهتمين بهذا الحدث. الشكل (8).

الشكل (8) خدمة تسليم الحدث ممثلة بوسيط مركزي (8)

3-2 خدمة تسليم الحدث على شكل شبكة من الوسطاء تشكل شبكة ند للند Peer to Peer:

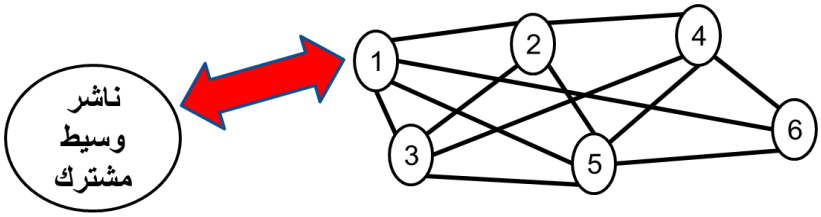
تتعاون مجموعة من الوسطاء في هذه الحالة لإنجاز المهام المتعلقة بتسجيل الاشتراكات وتسليم الأحداث المنشورة إلى المشتركين المهتمين بها. الشكل (9).



الشكل (9) خدمة تسليم الحدث ممثلة بشبكة من الوسطاء تشكل شبكة ند للند (Peer To Peer) (8)

3-3 خدمة تسليم الحدث يمكن أن تشمل أية عقدة في الشبكة:

كل عقدة في هذه الحالة يمكن أن تلعب دور الناشر أو المشترك أو الوسيط كما يبين الشكل (10).



الشكل (10) خدمة تسليم الحدث يمكن أن تشمل أية عقدة في الشبكة (8)

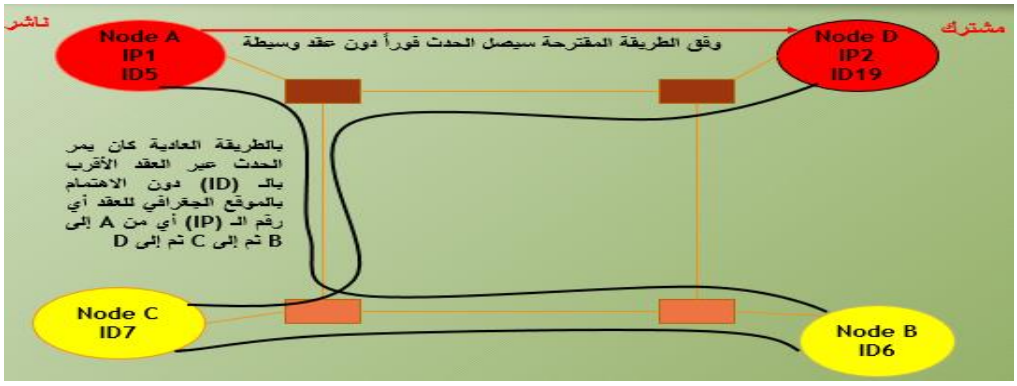
4- مواد البحث وطرائقه:

تم نمذجة الشبكة المقترحة والتي تستخدم بروتوكول PolderCast باستخدام برنامج PeerSim الخاص بنمذجة شبكات النشر والاشتراك ضمن بيئة eclipse البرمجية بحيث تكون:

- خدمة تسليم الحدث عبارة عن شبكة ند للند Peer to Peer.
- الطريقة المُستَخدَمة لتسجيل الاشتراك: اعتمادا على الموضوع Topic-Based.
- تتلخص الطريقة المقترحة بأخذ الموقع الجغرافي للعقد بعين الاعتبار، بالتالي فإن المجموعات التي تصلها الأحداث لا تتعلق بالعقدة ذات المعرفID فقط بل بالموقع الجغرافي لمجموعات العقد IP.

بفرض عقدتينA وD تقعان في موقع جغرافي واحد ولكن زمن انضمامهما للشبكة متباعد أي يملكان ID's بعيدة عن بعضها. فإنه عندما تريد إحداهما إرسال حدث والأخرى مهتمة به كان على الحدث المرور عبر العقدتين B وC أولاً لأنهما يملكان رقم ID متتالي أو قريب من بعضه، أما بعد التحسين المقترح فإن الحدث سوف يصل من A إلى D مباشرة دون المرور بأي عقد وسيطة كونهما ضمن نفس الموقع الجغرافي.

والشكل (11) يبين بشكل مبسط الآلية المقترحة:



الشكل (11) آلية الطريقة المقترحة

في بداية تهيئة الشبكة قمنا بإضافة عناوين ip للعقد وتأخذ الأرقام من:

0.0.0.0

انتهاءً بعنوان آخر عقدة حسب عدد العقد الكلي المستخدم في كل محاكاة ضمن الشبكة على حده.

تم اختيار العقد في بروتوكول vicinity حسب المواضيع المشتركة وحسب أولوية الموضوع أي كلما كانت المواضيع المشتركة بين العقدتين أكثر كان لها احتمالية أكبر أن تكون ضمن جدول العقدة. وأيضا كلما كانت أولوية المواضيع عالية كلما زاد احتمال أن تكون ضمن جدول العقدة.

- يقصد بزمن تهيئة الشبكة initialization time: عدد الدورات اللازمة لكي تبدأ الشبكة العمل بشكل مناسب، وخلال هذا الزمن لا يتم تبادل أي أحداث بين العقد. وتؤثر عدة عوامل على زمن تهيئة الشبكة كعدد العقد الموجودة والبروتوكول المستخدم لتبادل الأحداث بين العقد.
- تم تقييم زمن تهيئة الشبكة من خلال متوسط عدد الدورات cycles اللازمة لنشر واستقبال الأحداث. ويقصد بالدورة: عدد وحدات الزمن التي يلزم الحدث ليتم نشره واستقبال الرد بين عقدتين متجاورتين.

5- أدوات البحث:

Eclipse: هو بيئة تطوير متكاملة (IDE) ونظام لإضافة الملحقات (Plug-in) كُتب معظمه بلغة جافا (JAVA) ويمكن استخدامه لتطوير تطبيقات بلغة الجافا، أو بلغات البرمجة التي من بينها C, C++، وهو برنامج مفتوح المصدر ومجاني(11,12,13).

PeerSim: هو محاكي (simulator) مكتوب بلغة جافا ومفتوح المصدر ومجاني يستخدم ضمن بيئة الـ eclipse البرمجية.

تم تطويره مع قابلية عالية للتوسعية وديناميكية عالية (خروج وانضمام العقد من وإلى الشبكة) وهو يتكون من محركي محاكاة الأول مقاد بالدورة (الزمن) (cycle driven) والثاني مقاد بالحدث (event driven).

5-1 أنماط عمل المحاكي (PeerSim):

1- النمط المقاد بالدورة (الزمن) cycle driven:

هو محرك محاكاة (13,14) يعمل وفق دورات الساعة cycles، وهو ثابت تماماً fully static. وبالنسبة له فإن التهيئة يمكن أن تصف مجموعة من البروتوكولات وترتيبها، ومجموعة من التحكم وترتيبها، ومجموعة من المهيئات وترتيبها.

ينفذ هذا المحرك خارج مجموعة البروتوكولات تلك التي ترتبط فقط مع واجهة بروتوكول هذا النمط cycle driven protocol (CDProtocol).

تتم التجربة كالتالي:

يتم تشغيل أدوات التهيئة بالترتيب المحدد، ثم يتم تكرارها بعدد الدورات المحددة مسبقاً، وعند كل دورة ساعة يتم جدولة النتائج أي يتم تشغيل المحاكي وأخذ النتائج المطلوبة كل دورة ساعة.

2- النمط المقاد بالأحداث event driven:

هو محرك محاكاة يعمل وفق دورات الأحداث events، وهو ثابت تماماً fully static. وبالنسبة له فإن التهيئة يمكن أن تصف مجموعة من البروتوكولات وترتيبها، ومجموعة من التحكم وترتيبها، ومجموعة من المهيئات وترتيبها. ويعمل كالتالي:

يتم تشغيل أدوات التهيئة بالترتيب المحدد، ثم يجدول أول تنفيذ لكل عناصر التحكم المحددة في رتل الأحداث event queue، ثم يتم أخذ أول حدث من رتل الأحداث يغطي التحكم ويتم تنفيذه، وإلا يتم توصيل الحدث إلى البروتوكول الهدف المتضمن في بروتوكول Event Driven Protocol (EDProtocol). وهذه العملية تتكرر حتى انتهاء زمن المحاكاة أو عندما يصبح رتل الأحداث فارغاً.

إذا وقع أكثر من حدث في نفس اللحظة الزمنية فإنه يتم أخذ ترتيب الأحداث في الرتل بعين الاعتبار. أما إذا لم تكن الأحداث تحكمية ووقعت في نفس اللحظة يتم أخذها بترتيب عشوائي.

هذا المحرك أيضاً يؤمن الواجهة لإضافة أحداث للرتل. ونلاحظ أن هذا المحرك لا ينفّذ البروتوكول بشكل صريح.

في كافة الأحوال، يجب أن يعرّف عنصر تحكمي واحد أو عنصر تهيئة لإرسال الأحداث للبروتوكولات. يمكن جدولة الأحداث باستخدام بارامترات التهيئة لتعمل بعد انتهاء التجربة.

عندما ترسل البروتوكولات رسائل في عقد مختلفة لبعضها يمكنها استخدام نموذج طبقة النقل transport layer وبالتالي يتم في المحاكاة الأخذ بعين الاعتبار لتأخيرات الرسائل delay أو ضياع الرسائلOmissions .

6-النتائج العملية:

قمنا بمحاكاة بروتوكول PolderCast ضمن محاكي PeerSim بالإضافة لمحاكاة التعديل الذي تم إجراؤه وإيجاد مجموعة من النتائج التجريبية لتقييم أداء شبكة النشر والاشتراك ومعرفة مدى التحسين الذي تم في الحالتين التاليتين:

1- الحالة الاولى: قبل تطبيق التحسين على بروتوكول Polder Cast.
2- الحالة الثانية: بعد تطبيق التحسين المقترح على أداء عمل البروتوكول.

قمنا بتغيير عدد العقد وعدد المواضيع في الشبكة عند كل تجربة للوقوف بشكل أفضل على مدى فعالية التحسين بالنسبة لعدة جوانب في شبكات النشر والاشتراك.

1-6 التجربة الأولى:

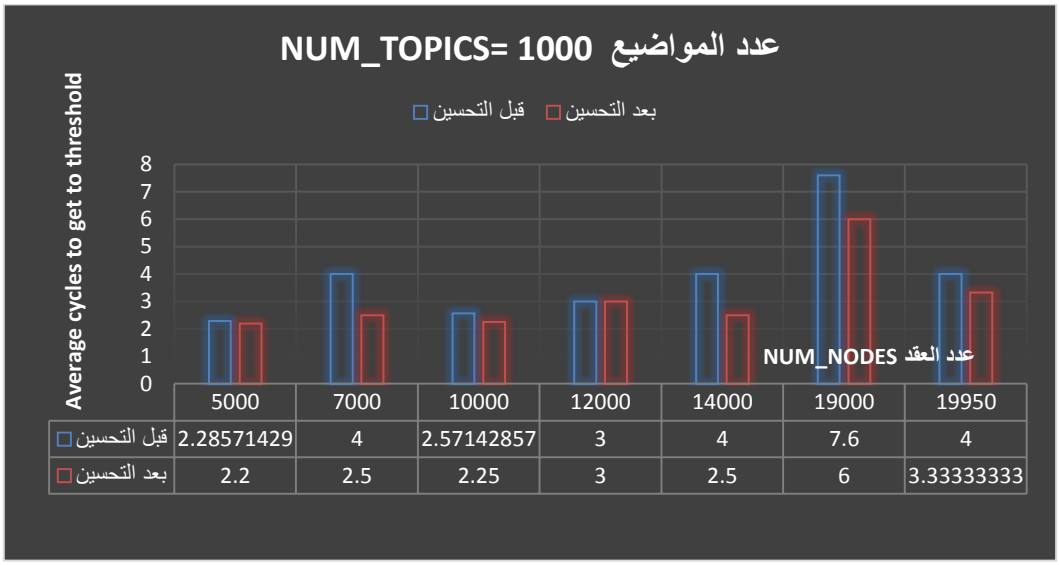
تم قياس العدد الوسطي للدورات (Cycles) لكي تصل الشبكة إلى الـ Threshold، ويقصد بهذا المصطلح عدد الدورات اللازمة لكي تصل الأحداث إلى 80% من عقد الشبكة.

في هذه التجربة تم اختيار عشرة عقد من الشبكة، الخمسة عقد الأولى للمواضيع الأكثر اشتراكاً بين العقد، والخمسة الأخرى للمواضيع الأقل اشتراكاً بين عقد الشبكة.

قمنا بتكرار التجربة على عدد مختلف من العقد ومن المواضيع، وكررنا التجربة بعد تطبيق التحسين على الشبكة فكانت النتائج كالتالي:

الاختبار الأول:

في هذا الاختبار كان عدد المواضيع NUM_TOPICS = 1000 موضوع، بينما تنوع عدد عقد الشبكة كما هو موضح في الشكل (12):

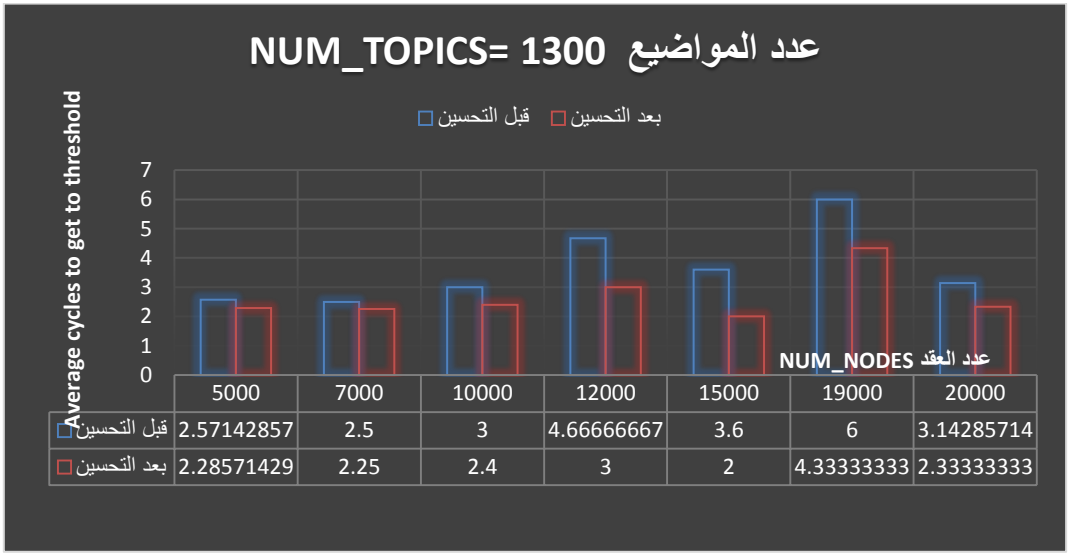


	5000	7000	10000	12000	14000	19000	19950
قبل التحسين	2.28571429	4	2.57142857	3	4	7.6	4
بعد التحسين	2.2	2.5	2.25	3	2.5	6	3.33333333

الشكل (12) عدد الدورات الوسطي للوصول للحد قبل وبعد التحسين عندما كان عدد المواضيع 1000 موضوع

الاختبار الثاني:

في هذا الاختبار كان عدد المواضيع NUM_TOPICS = 1300 موضوع، بينما تنوع عدد عقد الشبكة كما هو موضح في الشكل (13):

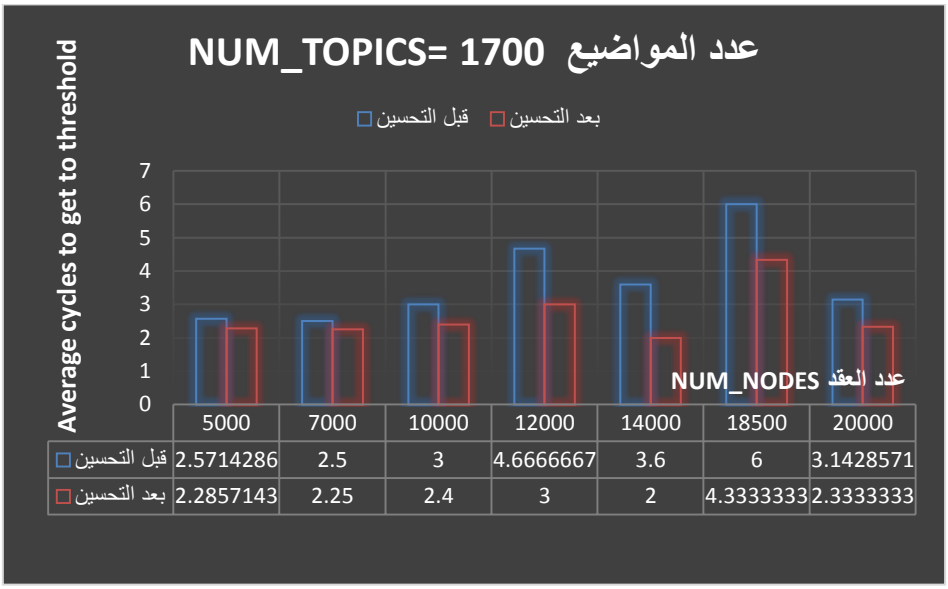


	5000	7000	10000	12000	15000	19000	20000
قبل التحسين	2.57142857	2.5	3	4.66666667	3.6	6	3.14285714
بعد التحسين	2.28571429	2.25	2.4	3	2	4.33333333	2.33333333

الشكل (13) عدد الدورات الوسطي للوصول للحد قبل وبعد التحسين عندما كان عدد المواضيع 1300 موضوع

الاختبار الثالث:

في هذا الاختبار كان عدد المواضيع NUM_TOPICS = 1700 موضوع، بينما تنوع عدد عقد الشبكة كما هو موضح في الشكل (14):

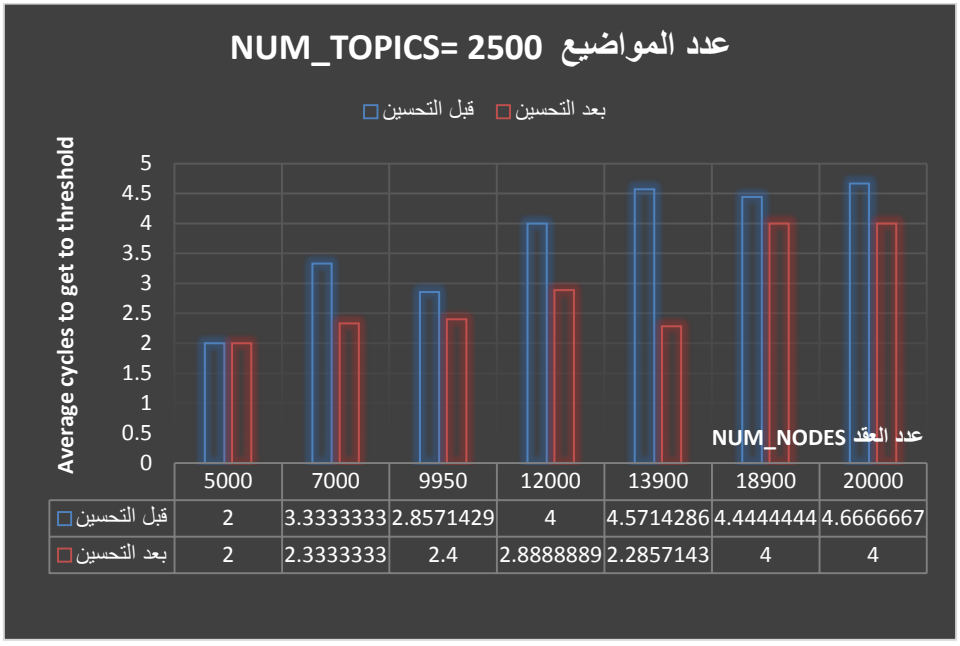


	5000	7000	10000	12000	14000	18500	20000
قبل التحسين	2.5714286	2.5	3	4.6666667	3.6	6	3.1428571
بعد التحسين	2.2857143	2.25	2.4	3	2	4.3333333	2.3333333

الشكل (14) عدد الدورات الوسطي للوصول للحد قبل وبعد التحسين عندما كان عدد المواضيع 1700 موضوع

الاختبار الرابع:

في هذا الاختبار كان عدد المواضيع NUM_TOPICS = 2500 موضوع، بينما تنوع عدد عقد الشبكة كما هو موضح في الشكل (15):

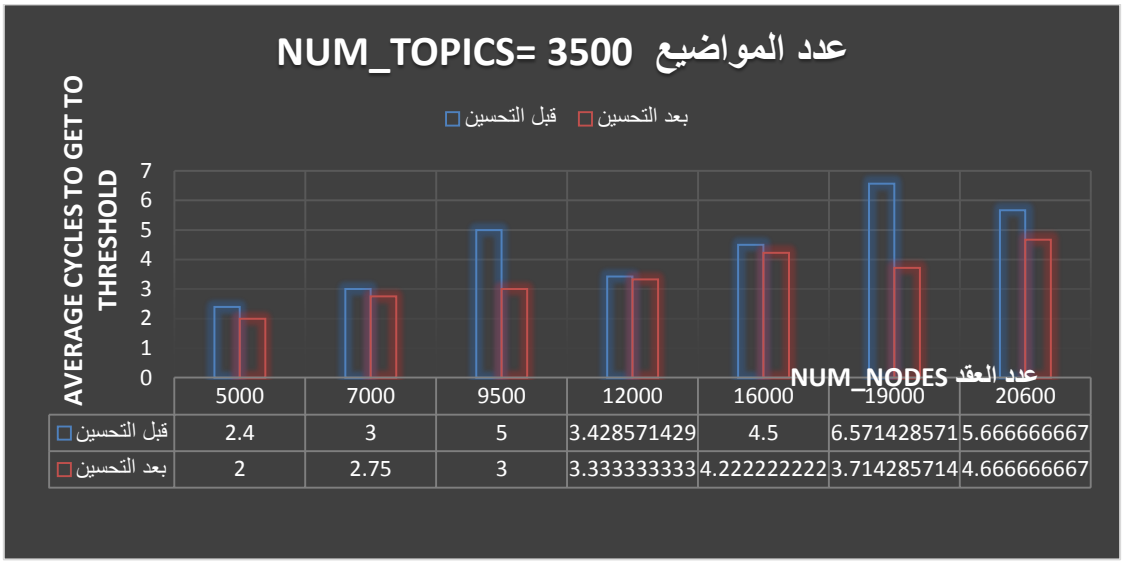


	5000	7000	9950	12000	13900	18900	20000
قبل التحسين	2	3.3333333	2.8571429	4	4.5714286	4.4444444	4.6666667
بعد التحسين	2	2.3333333	2.4	2.8888889	2.2857143	4	4

الشكل (15) عدد الدورات الوسطي للوصول للحد قبل وبعد التحسين عندما كان عدد المواضيع 2500 موضوع

الاختبار الخامس:

في هذا الاختبار كان عدد المواضيع NUM_TOPICS = 3500 موضوع، بينما تنوع عدد عقد الشبكة كما هو موضح في الشكل (16):



	5000	7000	9500	12000	16000	19000	20600
قبل التحسين	2.4	3	5	3.428571429	4.5	6.571428571	5.666666667
بعد التحسين	2	2.75	3	3.333333333	4.222222222	3.714285714	4.666666667

الشكل (16) عدد الدورات الوسطي للوصول للحد قبل وبعد التحسين عندما كان عدد المواضيع 3500 موضوع

نلاحظ من النتائج السابقة بأن عدد الدورات الوسطي للوصول لـ Threshold قبل التحسين أكبر في جميع الاختبارات التي أجريت مما هي عليه بعد إجراء التحسين.

2-6 التجربة الثانية:

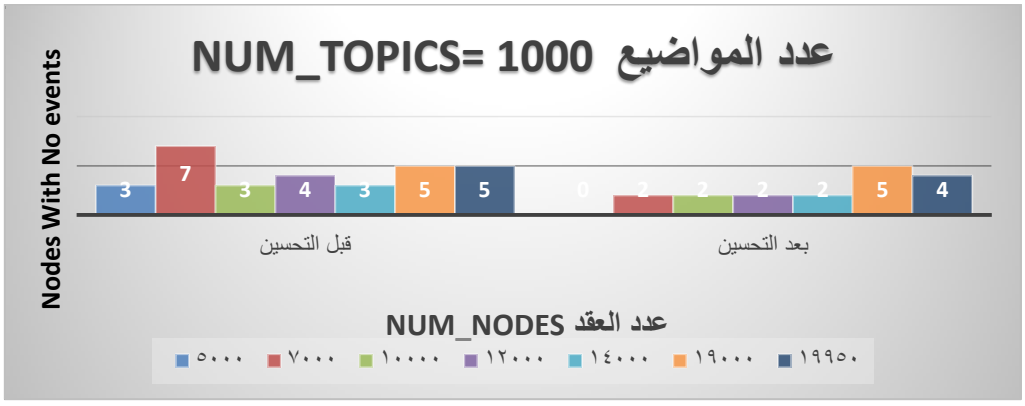
في هذه التجربة تم اختيار عشرة عقد عشوائية من الشبكة، بحيث تشترك كل عقدة من الخمسة الأولى بموضوع من الخمسة المواضيع الأكثر اشتراكاً بين العقد، أما الخمسة عقد الأخرى فتشترك كل عقدة منها بموضوع من الخمسة مواضيع الأقل اشتراكاً بين عقد الشبكة، وقمنا بإحصاء عدد العقد التي لم يصلها أحداث حول الموضوع المشتركة به خلال زمن تنفيذ التجربة.

قمنا بتكرار التجربة على عدد مختلف من العقد ومن المواضيع، وكررنا التجربة بعد تطبيق التحسين على الشبكة فكانت النتائج كالتالي:

الاختبار الأول:

في هذا الاختبار كان عدد المواضيع NUM_TOPICS = 1000 موضوع، بينما تنوع عدد عقد الشبكة كما هو موضح في الشكل (17):

النتائج المبينة في المخطط هي من أصل 10، كون عدد العقد التي اختبرناها في التجربة هو عشرة عقد.

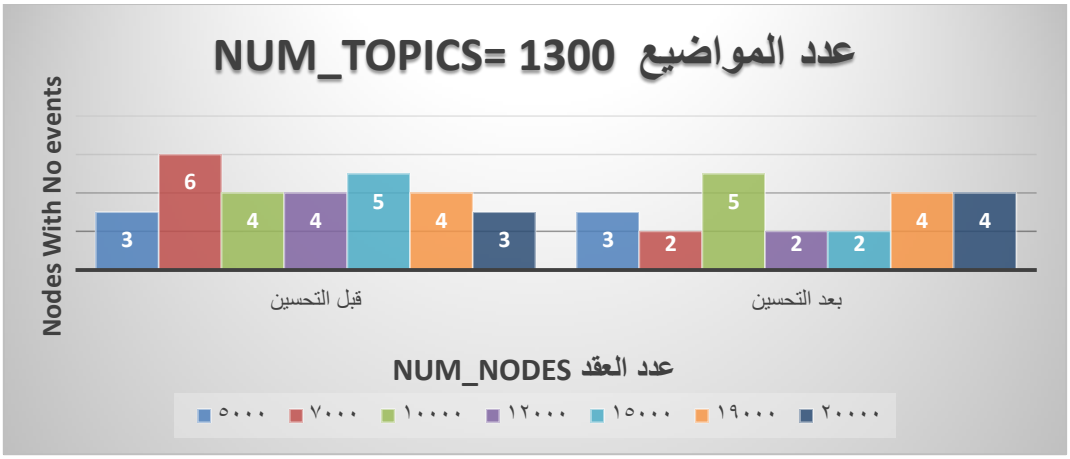


الشكل (17) عدد العقد التي لم تصلها أحداث قبل وبعد التحسين عندما كان عدد المواضيع 1000 موضوع

الاختبار الثاني:

في هذا الاختبار كان عدد المواضيع NUM_TOPICS = 1300 موضوع، بينما تنوع عدد عقد الشبكة كما هو موضح في الشكل (18):

النتائج المبينة في المخطط هي من أصل 10، كون عدد العقد التي اختبرناها في التجربة هو عشرة عقد.

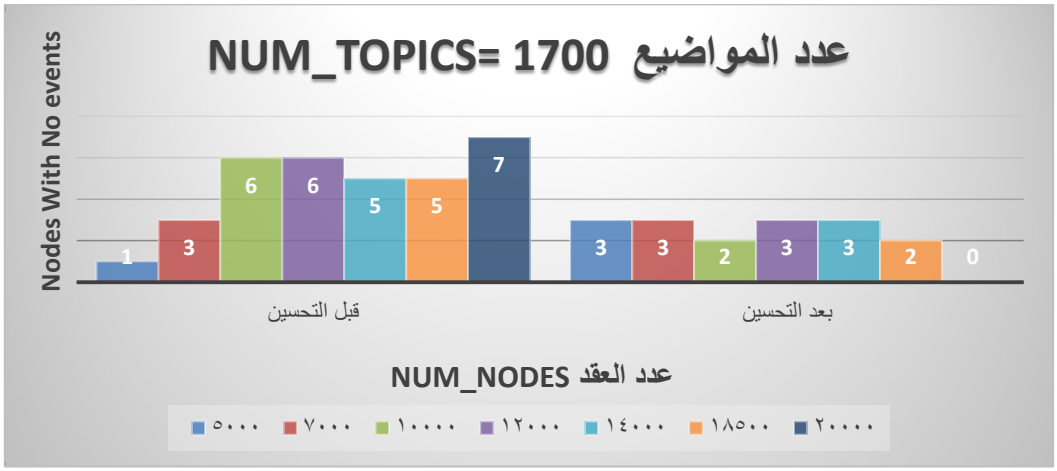


الشكل (18) عدد العقد التي لم تصلها أحداث قبل وبعد التحسين عندما كان عدد المواضيع 1300 موضوع

الاختبار الثالث:

في هذا الاختبار كان عدد المواضيع NUM_TOPICS = 1700 موضوع، بينما تنوع عدد عقد الشبكة كما هو موضح في الشكل (19):

النتائج المبينة في المخطط هي من أصل 10، كون عدد العقد التي اختبرناها في التجربة هو عشرة عقد.

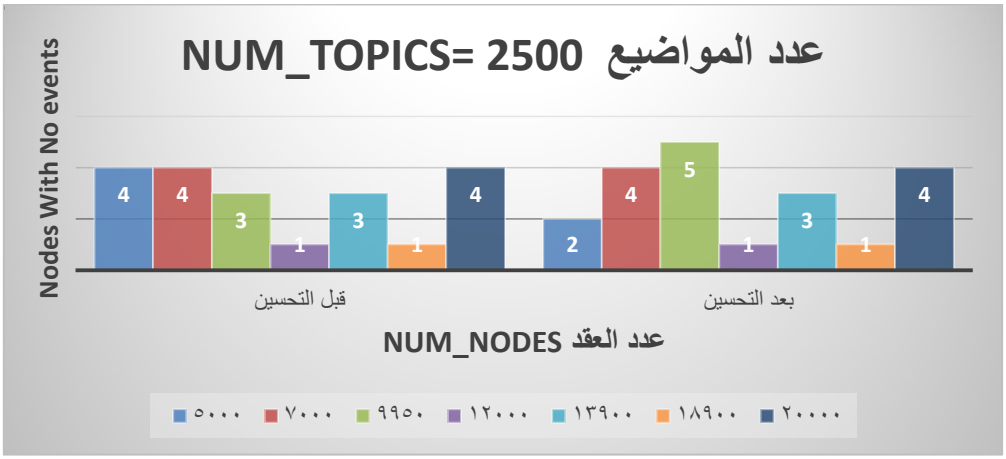


الشكل (19) عدد العقد التي لم تصلها أحداث قبل وبعد التحسين عندما كان عدد المواضيع 1700 موضوع

الاختبار الرابع:

في هذا الاختبار كان عدد المواضيع NUM_TOPICS = 2500 موضوع، بينما تنوع عدد عقد الشبكة كما هو موضح في الشكل (20):

النتائج المبينة في المخطط هي من أصل 10، كون عدد العقد التي اختبرناها في التجربة هو عشرة عقد.

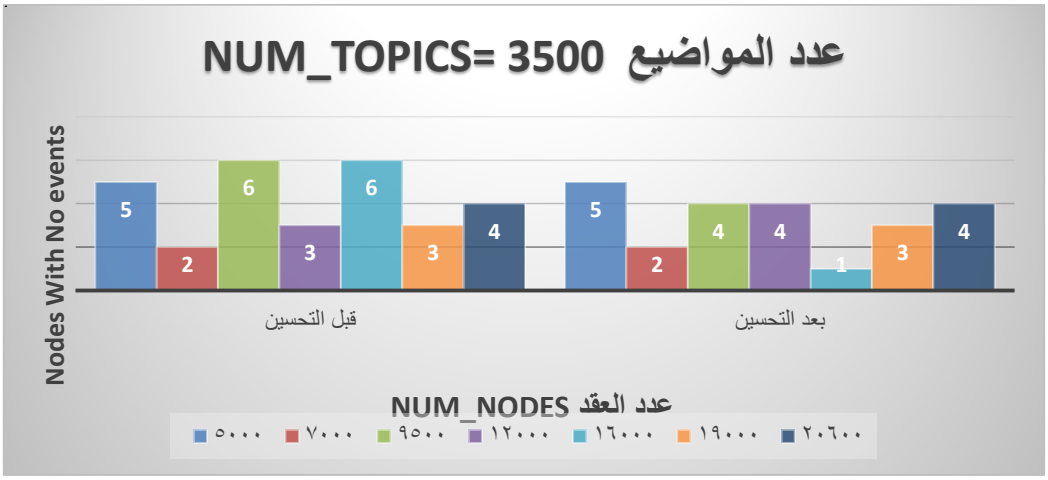


الشكل (20) عدد العقد التي لم تصلها أحداث قبل وبعد التحسين عندما كان عدد المواضيع 2500 موضوع

الاختبار الخامس:

في هذا الاختبار كان عدد المواضيع NUM_TOPICS = 3500 موضوع، بينما تنوع عدد عقد الشبكة كما هو موضح في الشكل (21):

النتائج المبينة في المخطط هي من أصل 10، كون عدد العقد التي اختبرناها في التجربة هو عشرة عقد.



الشكل (21) عدد العقد التي لم تصلها أحداث قبل وبعد التحسين عندما كان عدد المواضيع 3500 موضوع

نلاحظ من النتائج السابقة بأن عدد العقد التي لم يصل لها أي حدث حول المواضيع المقترحة المشتركة بها قبل التحسين أكبر في معظم الاختبارات التي أجريت مما هي عليه بعد إجراء التحسين.

7- المناقشة:

1- تم تصميم شبكة نشر واشتراك وتحليل أدائها من خلال برنامج PeerSim.

2- تم وضع خوارزمية لعمل بروتوكول التقارب في الشبكة تعتمد على أخذ الموقع الجغرافي بعين الاعتبار.

3- تبين من خلال التجارب بأن العدد الوسطي للدورات (Cycles) لكي تصل الشبكة إلى الـ Threshold كان أكبر قبل التحسين المقترح مما هو عليه بعد التحسين في جميع الاختبارات التي أجريت على أرقام مختلفة لعدد العقد والمواضيع.

4- تم تحسين زمن وصول الأحداث للمشتركين بمقدار 51.11% مقارنة بالتصميم الأصلي للبروتوكول (9).

5- عند اختيار عشرة عقد عشوائية من الشبكة، بحيث تشترك كل عقدة من الخمسة الأولى بموضوع من الخمسة المواضيع الأكثر اشتراكاً بين العقد، أما الخمسة عقد الأخرى فتشترك كل عقدة منها بموضوع من الخمسة مواضيع الأقل اشتراكاً بين عقد الشبكة. وعند إحصاء عدد العقد التي لم يصلها أحداث حول الموضوع المشتركة به خلال زمن تنفيذ التجربة لوحظ بأن عدد تلك العقد التي لم يصل لها أي حدث حول المواضيع المقترحة المشتركة بها قبل التحسين أكبر في معظم الاختبارات التي أجريت مما هي عليه بعد إجراء التحسين وقد بلغت نسبة التحسين 20%.

نعتقد بأن تقليل عدد العقد التي لم يصلها أحداث حول الموضوع المشتركة به خلال زمن تنفيذ التجربة يلمس بشكل مباشر بارامتر مهم جداً ألا وهو موثوقية البروتوكول والذي يعد تحسينه حجر الأساس لتقييم فعالية أي بروتوكول. كما أن تحسين زمن وصول الأحداث للمشتركين يؤدي بشكل مباشر لرفع أداء وفعالية الشبكة بشكل كبير. وهذا ما كنا نهدف إليه في هذه الدراسة.

8- الخلاصة والتوصيات:

إن التقليل من زمن وصول الأحداث للمشتركين والتقليل من نسبة فقد الأحداث في شبكات النشر والاشتراك قد يفتح المجال واسعاً أمام تطوير مثل هذه الشبكات وهو ذو أهمية بالغة بسبب أهمية تلك الشبكات التي باتت تشغل حيزاً واسعاً من الاستخدامات اليومية للمستخدمين ومن عرض الحزمة المستخدم على الشبكة بشكل عام.

تم من خلال تطبيق التعديل المقترح على شبكات النشر والاشتراك التحسين من عدة نواحي على أداء تلك الشبكات وهذا ما بينته التجارب والاختبارات. وفيما يلي بعض التوصيات:

1- تطبيق التقنية المقترحة على أرض الواقع من خلال إحدى وسائل التواصل الاجتماعي مثلاً.
2- مقارنة نتائج التطبيق مع النتائج التجريبية التي حصلنا عليها لبيان مدى فعالية التطوير على أرض الواقع.
3- العمل على إيجاد مجالات أخرى لتطوير وتحسين شبكات النشر والاشتراك.
4- تحسين الأمن في هذه الشبكات ودراسة الاختراقات الممكنة.
5- دراسة الآثار الاجتماعية والاقتصادية لاستخدام هذه الشبكات.

9- قائمة المراجع:

[1] ALVEIRINHO.J; PAIVA.J- *Flexible and Efficient Resource Location in Large-Scale Systems.* The 4th ACM SIGOPS/ SIGACT Workshop on Large Scale Distributed Systems and Middleware.2010.

[2] COULOURIS.G; DOLLIMORE.J; KINDBERG.T; and BLAIR.G- *DISTRIBUTED SYSTEMS Concepts and Design Fifth Edition* - Addison Wesley, ISBN 0-13-214301-1. 2011.

[3] EDWARDS.L- *To what extent is PolderCast more effective than SCRIBE as a peer-to-peer publish-subscribe overlay network design?* - Computer Science department, 18 November, 2014.

[4] GRESELIN.F, ZITIKIS.R- *From the Classical Gini Index of Income Inequality to a New Zenga-Type Relative Measure of Risk: A Modeller's Perspective*. MDPI magazine -econometrics section, 25 January. 2018.

[5] PLATANIA.M - *Ordering, Timeliness and Reliability for Publish/ Subscribe Systems over WAN*- PhD thesis in Computer Engineering, Sapienza University of Roma, Italy. ,2011.

[6] SETTY.V- Pblish/ Subscribe for Large-Scale Social Interaction: Design, Analysis and Resource Provisioning. *PhD Thesis, the Faculty of Mathematics and Natural Sciences*, University of Oslo, No. 1595. p.p. 10 -45. 2015.

[7] RAHIMIAN.F, GIRDZIJAUSKAS.S, PAYBERAH.A.H and HARIDI.S- *Vitis: A Gossip-based Hybrid Overlay for Internet-scale Publish/ Subscribe Enabling Rendezvous Routing in Unstructured Overlay Networks*. Swedish Institute of Computer Science (SICS), Stockholm, p.p. 746 - 757 Sweden, 16-20 May. 2011.

[8] SALVADOR.Z- Client Mobility Support and Communication Efficiency in Distributed Publish/ Subscribe- *PhD thesis in Computer Science*, The University of Basque Country,Spain,2012.

[9] SETTY.V; STEEN.M; VITENBERG.R; VOULGARIS.S- *PolderCast: Fast, Robust, and Scalable Architecture for P2P Topic-based Pub/ Sub.* Proceedings of the 13th International Middleware Conference. Springer-Verlag. New York. pp. 271-291. 2012.

[10] Hasan. S and Curry. E ،2017 -Thematic Event Processing Dataset -Available online: https:/ / www.researchgate.net/ publication/ 263673956_Thematic_event_processing_dataset (accessed on 28 February 2017).

[11] Hasan. S. 2016 - **Loose Coupling in Heterogeneous Event-Based Systems via Approximate Semantic Matching and Dynamic Enrichment**. Ph.D. Thesis, National University of Ireland Galway, Galway, Ireland, 2016.

[12] Apache Lucene- **Welcome to Apache Lucene**. Available online: https:/ / lucene.apache.org/ (accessed on5 March 2018).

[13] Belguith. S, Cui. S, Asghar. M and Giovanni Russello. G, 17 June 2018, **Secure Publish and Subscribe Systems with Efficient Revocation,** Cyber Security Foundry, The University of Auckland, New Zealand.

[14] Netti. A, April 26, 2018, **An Introduction to PeerSim,** University of Bologna.