

تقييم ومقارنة أداء التحسينات على خوارزمية التقطير في انترنت الأشياء

د. محسن عبود

م. منال العمر

الملخص

اكتسبت خوارزمية Trickle التي تم تقديمها وتوحيدها في RFC 6206 شعبية كبيرة حيث يمكنها أن تضمن حل تناقض البيانات في البيئات الموزعة بدقة وسرعة مع انخفاض تكلفة الصيانة ودعم جيد للتوسع وزيادة كثافة الشبكة. بالنسبة لهذه الميزات الجذابة تشكل Trickle أساس العديد من معايير الإنترنت ويتم نشرها في العديد من التطبيقات مثل البث الموثوق واكتشاف الخدمة والموارد الموزعة وكذلك بروتوكولات التوجيه ومنها بروتوكول RPL المستخدم في شبكات انترنت الأشياء. قمنا في هذا البحث بالمقارنة بين أهم الدراسات التي حسنت خوارزمية Trickle من أجل تقليل التكلفة اللازمة لحل التناقض في معلومات التوجيه في بروتوكول RPL ، حيث قارنا بينهم من حيث زمن التقارب وعدد حزم التحكم المرسل في الشبكة و مقدار استهلاك الطاقة ومعدل تسليم الحزم وزمن تأخيرها. لقد وجدنا أن كل خوارزمية حسنت مجموعة من معايير الأداء وأن الخوارزميات التي قللت من زمن التقارب زادت من عدد الحزم التحكم وبالعكس، كما وجدنا أن خوارزمية Adaptive-k أفضل من غيرها كونها قللت عدد حزم التحكم وتأخيرها ومقدار الطاقة المستخدمة.

الكلمات المفتاحية: خوارزمية التقطير، انترنت الأشياء، بروتوكول RPL ، نظم موزعة.

Evaluation and comparing the performance of improvements to the IoT Trickle algorithm

Eng.Manal Alomar

Dr.Mohssen Abboud

Abstract

The Trickle algorithm introduced and standardized in RFC 6206 has gained great popularity as it can ensure data inconsistency resolution in distributed environments accurately and quickly with low maintenance cost and good support for scaling and increasing network density. For these attractive features, Trickle forms the basis of many Internet standards and is deployed in many applications such as reliable broadcasting, service discovery, distributed resources, as well as routing protocols, including the RPL protocol used in Internet of Things networks. In this paper, we compared the most important studies that improved the Trickle algorithm in order to reduce the cost required to solve the discrepancy in routing information in the RPL protocol. We compared them in terms of convergence time, number of control packets sent in the network, amount of energy consumption, packet delivery rate and latency time. We found that each algorithm improved a set of performance parameters, and the algorithms that reduced the convergence time increased the number of control packets and vice versa, and we also found that the Adaptive-k algorithm is better than others as it reduces the number of control packets, their latency, and the amount of energy used.

KEYWORDS: Trickle algorithm, Internet of Things, RPL protocol, distributed system.

1. المقدمة:

تتطلب شبكات LLN (Low-Power and Lossy Networks) بناء طوبولوجيا الشبكة بأسلوب يتسم بالكفاءة والسرعة، وبالتالي من الضروري وجود خطة فعالة لاختيار اتجاه البيانات والتوجيه بشكل سريع. يعتمد RPL (IPv6 Routing Protocol for LLN) وهو بروتوكول التوجيه في شبكات LLN من أجل الحفاظ على الطاقة والحفاظ على طوبولوجيا الشبكة مع أقل تكلفة للتوجيه خوارزمية Trickle التي تضمن الانتشار السريع لمعلومات التوجيه وتكلفة صيانة منخفضة لطوبولوجيا الشبكة. تستخدم هذه الخوارزمية أسلوب يسمى بالثرثرة المهذبة "polite gossip" وتعني أن المشارك لن ينشر ثرثرة إذا قام شخص آخر بذلك بالفعل وبالتالي بواسطة خوارزمية Trickle يتم التنظيم الذاتي لتبادل الحزم مع العقد المجاورة [2]. سنقوم في هذا البحث بتقييم أداء أهم التحسينات على خوارزمية Trickle في انترنت الأشياء.

2. مشكلة البحث:

على الرغم من الجهود البحثية الموجودة لتحسين خوارزمية Trickle إلا أنه لا توجد دراسات شاملة تعمل على تقييم أداء هذه الخوارزميات المقترحة ومقارنتها مع بعضها البعض لتحديد ميزات وعيوب كل منها وتحديد أفضلها.

3. الهدف من البحث:

يهدف هذا البحث إلى تحقيق برمجي لأغلب خوارزميات Trickle الموجودة في الدراسات وذلك من خلال فهم الكود البرمجي الخاص ببروتوكول RPL و كيفية تحقيق خوارزمية Trickle داخله، ثم التعديل على الملفات الخاصة بخوارزمية Trickle ضمن بروتوكول RPL ومن ثم تقييم أداء هذه الخوارزميات والمقارنة فيما بينها.

4. أهمية البحث:

نظرًا لحقيقة أن بروتوكول التوجيه هو أحد الأعمدة الرئيسية لهندسة الشبكات، ويتوقع بثقة ضرورة وجود شبكات LLN، أصبح RPL سريعًا بروتوكول التوجيه الفعلي لـ IoT، علاوة على ذلك لأن RPL هو بروتوكول التوجيه الموحد الوحيد لشبكات LLN حتى الآن، فإن عدد الأعمال المنشورة حول RPL يزداد كل عام بشكل واضح ومن أهمية خوارزمية Trickle في ضبط انتشار بيانات توجيه بروتوكول RPL تتبع أهمية بحثنا هذا الذي يقدم دراسة شاملة لأهم التحسينات على هذه الخوارزمية وتقييمها ومقارنتها وتقديم الاقتراحات للمهتمين في هذا المجال.

5. انترنت الأشياء:

يعرف انترنت الأشياء (IoT) بأنه شبكة معقدة تربط مليارات الأجهزة والبشر في بنية تحتية متعددة التقنيات ومتعددة البروتوكولات، تتمثل الرؤية الرئيسية لإنترنت الأشياء في خلق عالم ذكي يوفر مزيدًا من الذكاء للطاقة والصحة والنقل والمدن والصناعة والمباني والعديد من المجالات الأخرى، سيتم تحقيق ذلك من خلال جعل الكائنات التي نتعامل معها يوميًا مزودة بأجهزة استشعار ومحدد هوية ومحدد مواقع ولها عنوان IP لتصبح كائنات ذكية وقادرة على التواصل ليس فقط مع الكائنات الذكية الأخرى ولكن أيضًا مع البشر [3].

6. شبكات الخسارة والطاقة المنخفضة Low-Power and Lossy Networks (LLNs) [5]:

تعتبر LLN تمكين الشبكات اللاسلكية لإنترنت الأشياء (IoT) وهي واحدة من لبنات بناء انترنت الأشياء تتألف LLN من مجموعة من الأجهزة المدمجة المترابطة والتي تتميز عادة بالقيود على موارد العقدة وتقنيات الاتصال الأساسية [6]. يعتبر استخدام بروتوكول IP كبروتوكول أساسي لـ LLN ذو فوائد واضحة. حيث يمكن هذه الأجهزة من الاتصال

بالإنترنت ويسمح أيضاً لهذه الأجهزة بالاستفادة من الواجهات القياسية المستندة إلى الويب وأدوات الإدارة القياسية ومع ذلك لم يتم تصميم بروتوكولات TCP / IP لدعم الأجهزة منخفضة الطاقة التي تعمل في بيئة شديدة الخسارة لذلك هناك مجموعة واسعة من البروتوكولات اللاسلكية المختلفة المستخدمة لتوصيل هذه الأجهزة بالإنترنت ومن أهم البروتوكولات التي تمكن IP من العمل بفعالية في بيئة محدودة الموارد ومتعرضة للخسارة نذكر (6LoWPAN) IPv6 over Low power WPAN's و IPv6 Routing Protocol for LLNs(RPL) [7].

6-1 . بروتوكول توجيه IPv6 للشبكات منخفضة الطاقة والخسارة (RPL) [2]:
هو بروتوكول توجيه IPv6 لشبكات LLNs حيث تم اقتراحه للشبكات ذات الوصلات المفقودة التي تكشف عن معدل عالي في أخطاء الحزم وبالإضافة إلى فصل وانقطاع الوصلات. يعتبر بروتوكول RPL من النوع Source Routing (path addressing) حيث يسمح لمُرسل الحزمة بتحديد المسار الذي تتبعه الحزمة عبر الشبكة جزئياً أو كلياً ويتيح للعقدة اكتشاف جميع الطرق الممكنة للهدف كما يوفر RPL دعم التوجيه لنقل البيانات إما multicast أو unicast. تنظم RPL طوبولوجيا الشبكة على شكل DAG (Directed Acyclic Graph) الذي يتم تقسيمه إلى واحد أو أكثر من DODAG (Destination Oriented DAGs) حيث يتم تحديد المسارات الافتراضية بين العقد في LLN بواسطة هذه البنية الشجرية. يمكن أن تنتمي العقدة الموجودة في DAG إلى أكثر من والد مقارنة بالشجرة التقليدية. ترتبط العقد في بنية DODAG ليس فقط بوالدها ولكن أيضاً مع العقد الأخوية [8]. يكون جذر هذه الشجرة في عقدة المصرف ويعمل هذا الجذر كنقطة عبور إلى شبكات IPv6 ويتميز بكونه عقدة لا تحتوي على نفس القيود مثل عقدة المستشعر ويسمى جهاز التوجيه الحدودي. علاوة

على ذلك يتم التحكم في إرسال التحديثات إلى العقد الأخرى بواسطة جهاز التوجيه الحدودي.

6-1-1 . رسائل التحكم في RPL [8]:

(a) **IDODAG Information Solicitation (DIS)**: يتم إرسال هذه الرسالة من خلال عقد جديدة للانضمام إلى DODAG تلتزم من خلالها كائن المعلومات (DIO) من عقدة RPL.

(b) **DODAG Information Object (DIO)**: رسالة بث مرسله بواسطة عقدة الجذر لإنشاء DODAG يحتوي DIO على معلومات عامة مثل RPLInstanceID أو الترتيب أو DODAGID ، ، إلخ.

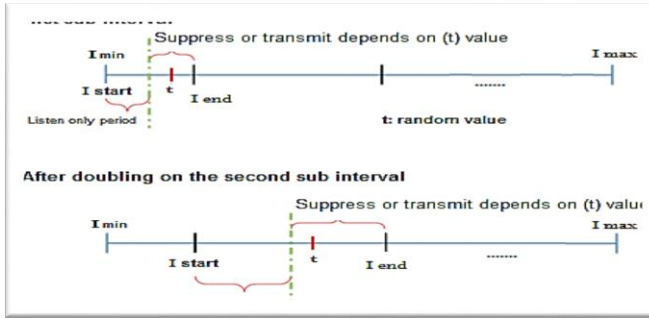
(c) **Destination Advertisement Object (DAO)**: إنها رسالة تُستخدم لإنشاء مسارات توجيه للأسفل من جذر DODAG إلى العقد الأخرى.

(d) **Destination Advertisement Object (DAO-ACK)**: ترسل هذه الرسالة من قبل مستلم رسالة DAO (جذر الـ DODAG) كرد على رسالة DAO أحادية الإرسال المستلمة. تحتوي على البيانات المتعلقة بالحالة، تسلسل DAO، ومعرف حالة الـ RPL .

6-1-2 . خوارزمية Trickle [9]:

يتكون RPL من مجموعة من الخوارزميات، لكل خوارزمية مهام محددة. الخوارزمية الرئيسية في RPL هي خوارزمية Trickle Timer. يتمثل الهدف الرئيسي لخوارزمية Trickle Timer في إدارة عملية الإرسال في الشبكة، حيث تقوم Trickle بتقليل نشر الرسائل التي لا تحتاج إلى إعادة الإرسال مثل الرسائل المتكررة في الشبكة. يتم ذلك باستخدام آليتين الأولى تحدث عندما تظهر حالة عدم تناسق في الشبكة، تزيد الخوارزمية من معدل التحكم في الإشارة لتعود إلى الوضع التوافقي في الشبكة، والثانية تحدث عند تكرار نفس الرسالة في الشبكة والعقد لم تعد بحاجة

إليها لأنها متصلة بجيرانها تعمل عندها الخوارزمية على كبح ارسال الرسالة وهذا يساعد على تقليل الرسائل المنتشرة على الشبكة وتوفير الطاقة. تقوم خوارزمية Trickle بتعيين الفاصل الزمني الرئيسي لكل عقدة في الشبكة، ويبدأ هذا الفاصل الرئيسي من I_{min} وينتهي عند I_{max} حيث كل من I_{max} و I_{min} متغيرات. تقسم العقدة الفاصل الزمني الرئيسي الخاص بها إلى مجموعة من المناطق الفرعية، كل فاصل زمني يبدأ من I_{start} وينتهي في I_{end} و I_{end} و I_{start} هي متغيرات. يبدأ التنفيذ من الفترة الفرعية الأولى في العقدة، وعند انتهاء الفترة الفرعية الأولى يبدأ الفترة الفرعية التالية، وما إلى ذلك حتى ينتهي من جميع المناطق الفرعية عندما يصل المؤقت إلى قيمة I_{max} .



الشكل (1) خوارزمية Trickle القياسية لكل عقدة [9].

تحتوي خوارزمية Trickle القياسية على ثلاثة بارامترات أساسية:

- I_{max} : الحد الأقصى لحجم الفاصل الزمني يوصف I_{max} على أنه عدد من مضاعفات I_{min} .
 - I_{min} : الحد الأدنى لحجم الفاصل.
 - K : القيمة الحدية وتسمى أيضاً بثابت التكرار k .
- محلياً، تحافظ كل عقدة في الشبكة على مؤقت وثلاثة متغيرات:
- a : حجم الفاصل الحالي.

- C: عدد لعدد رسائل Trickle المستلمة أثناء الفاصل الحالي ويسمى عدد الاتساق.
- t: وقت الإرسال وهو عشوائي محدد داخل الفاصل الحالي (a و $1/2$). يمكن التعبير عن خوارزمية Trickle من خلال القواعد أو الخطوات الستة أدناه[1]:
 1. **الخطوة 1:** عندما يبدأ Trickle التنفيذ فإنه يختار (a) بشكل عشوائي وموحد من $[Imin, Imin*2Imax]$ ويبدأ الفاصل الأول.
 2. **الخطوة 2:** عند بدء فاصل زمني (a) تعيد Trickle إعادة تعيين c إلى 0 وتختار t بشكل عشوائي من النطاق $[1, 1/2]$.
 3. **الخطوة 3:** كلما سمعت عقدة انتقال متناسق مع بياناتها تزيد Trickle عدد الاتساق c.
 4. **الخطوة 4:** في الوقت t يرسل Trickle إذا كان $(c < k)$ فقط وخلاف ذلك يتم قمع الإرسال.
 5. **الخطوة 5:** عند انتهاء الفاصل الزمني a يضاعف Trickle طول الفاصل حتى الوقت المحدد بواسطة $Imax$ ثم يبدأ فاصل زمني جديد كما في الخطوة 2.
 6. **الخطوة 6:** إذا سمع Trickle انتقال غير متناسق بينما a أكبر من $Imin$ فإنه يصفر المؤقت. للقيام بذلك يقوم Trickle بجعل a يساوي $Imin$ ويبدأ فاصلاً جديداً كما في الخطوة 2، وإلا مثلاً a تساوي $Imin$ عند اكتشاف التناقض ولن يفعل Trickle شيئاً. يضمن اختيار t من النصف الثاني من الفاصل الزمني في الخطوة 2 فترة استماع تعادل فقط نصف الفاصل ويتم ذلك استجابة لمشكلة الاستماع القصير.

(Constant(k) وعدد العقد والحد الأدنى لطول الفاصل الزمني الأول (Imin) والحد الأقصى لطول الفاصل الزمني (Imax) ووسط الراديو المستخدم في عملية المحاكاة ووقت الإرسال (Transmission time) ووقت المحاكاة ، وقد لوحظ أنه عند انخفاض زمن تقارب الشبكة يزداد استهلاك الطاقة وعندما يزداد زمن تقارب الشبكة ينخفض استهلاك الطاقة. يتناقص وقت تقارب الشبكة مع انخفاض قيمة Imin لأنه مع انخفاض قيمة Imin يتم إرسال حزم DIO بسرعة ، مما يؤدي إلى مزيد من استهلاك الطاقة ويزيد وقت تقارب الشبكة مع ارتفاع قيمة Imin لأن حزم DIO يتم إرسالها بمعدل بطيء وبالتالي فإن ينخفض استهلاك الطاقة أيضاً. إجمالي عدد الحزم المرسله أقل في خوارزمية E-Trickle أما إجمالي عدد الحزم المرسله يتناسب طردياً مع استهلاك الطاقة. يتناقص تأخير الشبكة أيضاً مع زيادة قيم Imin لأنه بسبب القيم الكبيرة لـ Imin يتم إرسال الحزم بمعدل بطيء جداً وبالتالي تمنع الاصطدامات كما تمنع إعادة إرسال الحزم.

تتميز دراستنا هذه عن هذه الدراسات السابقة بأننا سنقارن بين تسع خوارزميات Trickle مع الأخذ بعين الاعتبار اختلاف عدد العقد وسنقارن الأداء ليس فقط من حيث زمن التقارب بل أيضاً من حيث التأخير واستهلاك الطاقة ومعدل تسليم الحزم وعدد حزم التحكم المرسله على الشبكة.

8. التحسينات على خوارزمية التقطير في RPL:

- **Trickle-F** : اقترح الباحث في [11] نسخة معدلة من خوارزمية Trickle وهي Trickle-F والتي تهدف إلى حل مشكلة موازنة التحميل من خلال ضمان آلية عادلة لكبح البث على المدى القصير بين العقد في المنطقة لتسهيل الاكتشاف السريع لجميع المسارات المتاحة. يتمثل الأساس المنطقي وراء Trickle-F في تحديد أولوية كل عقدة اعتماداً على عدد عمليات القمع المتتالية بمعنى أنه كلما

طال الوقت الذي تقضيه العقدة دون الإرسال زادت أولويتها للإرسال في الجولة التالية.

- **I-Trickle**: الخوارزمية المقترحة في [12] والتي تدعى I-Trickle تضع قيمة عداد التكرار redundancy counter عند قيمة الصفر ليس في بداية الفاصل الزمني الجديد ولكن في وقت منع أو إرسال رسالة كائن معلومات DODAG في خوارزمية Trickle من أجل حل مشكلة موازنة الحمل وتقليل استهلاك الطاقة من خلال مراعاة الرسائل التي يتم سماعها من وقت t إلى نهاية الفاصل الزمني.
- **Optimized Trickle**: تم الاقتراح في [1] تحسين بسيط وقوي يقلل من التأخير دون تكبد أي نفقات عامة إضافية. في هذه الخوارزمية المحسنة عندما يتم اكتشاف عدم تناسق، يعيد تعيين I إلى Imin الذي يتم أيضاً في خوارزمية Trickle الأصلية. لكنه يقول أنه في هذا الوقت سيختار الوقت العشوائي t في [0, Imin] بدلاً من [I/2, I] كما هو الحال في خوارزمية Trickle الأصلية.
- **E-Trickle**: في الورقة [14] تم اقتراح نسخة محسنة من Trickle وهي E-Trickle حيث تقدم هذه الخوارزمية ثلاث تعديلات لخوارزمية Trickle. أولاً بدلاً من تحديد الوقت العشوائي t من النطاق [I/2, I] يتم تحديد قيمة t من النطاق [0, I]. ثانياً بدلاً من تعيين عداد التكرار c على قيمة 0 في بداية كل فاصل زمني يتم تعيين c إلى قيمة 0 فقط في بداية الفاصل الزمني الأول Imin وأيضاً في الوقت الذي تم اختياره عشوائياً t وذلك من أجل القضاء على التأثير التراكمي لمشكلة الاستماع القصير. كما قد يلاحظ المرء، فإن ضبط عداد التكرار c إلى 0 في الوقت المحدد عشوائياً سيؤدي إلى فواصل زمنية غير متساوية بين العقد من حيث الطول وبالتالي، فإن العقد ذات طول الفاصل الأقل سيكون لها فرص أكبر للإرسال. للتغلب على هذا الموقف تم إضافة خطوة ثالثة لتعديل قيمة عامل التكرار k حيث تصبح قيمته $(newk = ((2 * \ln z - 1) / I) * k)$ بشكل متكيف لتعكس التمدد الذي يحدث في حجم الفاصل .

- **Adaptive-k**: يقترح الباحث في [15] امتداداً لخوارزمية Trickle التي تسمى adaptive-k والتي تسمح للعقد بتعيين ثابت التكرار الخاص بها وفقاً للمعلومات المحلية حول كثافة الشبكة. حيث تعتمد كل عقدة على العداد c لتخمين عدد جيرانها وتسمح للعقد بتعيين قيمتها k بشكل مستقل .
- **Trickle-Plus**: يوجد في [17] نسخة موسعة من خوارزمية Trickle. في هذه النسخة بدلاً من زيادة حجم نافذة الإرسال أضعافاً مضاعفة، يتم وضع حجم الفاصل الزمني الجديد بالقيمة $(I * SF * 2)$ حيث SF هو عامل الازاحة ويعني عدد مرات مضاعفة الفاصل الزمني التي يمكن تخطيها، وبالتالي تتقارب الشبكة بسرعة مع استهلاك أقل للطاقة.
- **RANDOMIZED DYNAMIC TRICKLE**: اقترح الباحث في [19] تحسيناً لخوارزمية Trickle القياسية مرتبطاً بقيمة t . في الخوارزمية المقترحة يتم تحديد وقت لفترة الاستماع وفترة الإرسال بناءً على قيمة t ضمن فترة زمنية فرعية. المتغير t هو وقت عشوائي يستخدم كمحدد للاستماع والإرسال. لذلك يركز هذا العمل بشكل أساسي على اختيار قيمة t . حيث يتم اختيار قيمة t وفقاً لكثافة الشبكة، لتحقيق ذلك يتم اقتراح أربع حالات لاختيار قيمة t أفضل في الفترة الفرعية. في كل مرة يتم التحقق من عدد الجيران (C) لتحديد النطاق الذي يجب أن يتم اختيار فيه وقتاً عشوائياً (t).
- **FL-Trickle**: قدم الباحثون في [21] تحسيناً جديداً لخوارزمية Trickle القياسية وهي خوارزمية Flexible Trickle (FL-Trickle) تقلل FL-Trickle من التأخير في إرسال رسائل التحكم عن طريق تحديد وقت الإرسال T عند $1/2$ بدلاً من اختياره بشكل عشوائي في $[1, 1/2]$. تقلل FL-Trickle من معدل النقل من خلال العمل بقيمة عالية من الحد الأدنى للفاصل الزمني (I_{min}) من أجل الحصول على عبء منخفض مع عدم وجود زيادة في استهلاك الطاقة.
- **Elastic Trickle**: تحاول الخوارزمية المرنة المقترحة في [22] التعامل مع مشكلة فترة الاستماع فقط حيث تم اكتشاف أن فترة الاستماع تأثرت بعدد الجيران.

تتمثل المساهمة الرئيسية لهذه الخوارزمية أنها تتيح الاختيار الديناميكي لفترة الاستماع فقط بناءً على عدد الجيران أو العقد المحيطة.

9. القسم العملي :

9-1. نظام التشغيل Contiki ومحاكي الشبكات Cooja [16]:

تم اختيار Contiki لأنه تم تصميمه خصيصاً لأجهزة IoT منخفضة الطاقة والخسارة ولديه تطبيق أساسي لخوارزمية Trickle في مكتبة ContikiRPL التي سيتم استخدامها كأساس لبحثنا . كما تم اختيار المحاكي Cooja الذي يعمل على نظام تشغيل Contiki (الإصدار 3.0). يعتمد اختيار هذا المحاكي على اعتبارين رئيسيين: الأول هو محاكي مفتوح المصدر مصمم لتطبيقات إنترنت الأشياء. ثانياً يجعل تعديل وتحسين خوارزمية Trickle سهلاً جداً بسبب تنفيذه على RPL الأساسي. كما يوفر محاكي Cooja أدوات لإخراج البيانات من كل اختبار بسهولة شديدة بتنسيق قابل للقراءة. هذا ما يسهل علينا استخراج البيانات منها وإنتاج الرسوم البيانية من المخرجات باستخدام نصوص Perl لتصفية البيانات الناتجة إلى مقاييس الأداء المرغوب فيها، بالإضافة إلى ذلك يوفر هذا المحاكي أداة تدعى collect view التي يمكن للمبرمج دمجها بالكود الخاص به وتعرض معلومات ومخططات مفيدة في عملية قياس الأداء.

9-2. المقاييس المستخدمة لتقييم الأداء ومقارنته:

- **The Convergence Time (setup time):** وقت تقارب الشبكة مقدار الوقت الذي تحتاجه جميع العقد القابلة للوصول (من حيث الراديو) في الشبكة للانضمام إلى DAG.

$$\text{Convergence Time} = \text{Last DIO joined DAG} - \text{First DIO sent}$$

- **Energy Consumption**: مقدار استهلاك الطاقة بالميلي واط من قبل كل من المعالج والاستماع والارسال ونمط توفير الطاقة LPM.
 - **Control Traffic Overhead**: وهذا يشمل رسائل DIO و DIS و DAO التي تم إنشاؤها بواسطة كل عقدة.
- $$Control\ Traffic\ Overhead = \sum_{k=1}^n DIO(k) + \sum_{k=1}^m DIS(k) + \sum_{k=1}^o DAO(k)$$
- **Packet Latency**: زمن تأخر الحزمة. يُعرف زمن التأخر بأنه مقدار الوقت الذي تستغرقه حزمة من العقدة للوصول إلى جهاز التوجيه الحدودي أو المصرف وهو متوسط زمن الوصول لجميع الحزم في الشبكة من جميع العقد.
- $$Total\ Latency = \sum_{k=1}^n (Recv\ Time(k) - Sent\ Time(k)).$$
- **Packet Delivery Ratio (PDR)**: ويتم تعريفه على أنه عدد الحزم المستلمة في المصرف إلى عدد الحزم المرسله إليه. نأخذ متوسط PDR لجميع الحزم التي تم استلامها بنجاح في المصرف.
- $$Average\ PDR = (Total\ Packets\ Received / Total\ Packets\ Sent) * 100$$

3-9. معاملات المحاكاة [23]:

قمنا بإجراء اختبارات مكثفة لتحديد مدة تشغيل المحاكاة بحيث يكون للقيم التي تم الحصول عليها تبايناً صغيراً وللتأكد من أن وقت إعداد الشبكة لم يؤثر بشكل كبير على النتائج. نظراً لأن الاختبارات التي تم إجراؤها لمدة 7 دقائق أسفرت عن نتائج مشابهة لتلك التي أجريت لمدة 20 دقيقة، فقد اخترنا 7 دقائق كوقت المحاكاة لجميع عمليات محاكاة Cooja. تحوي طوبولوجيا الشبكة على 20 ثم 40 عقدة عميل في منطقة 100 م × 100 م. تمثل العقدة 1 في الزاوية اليسرى العليا جهاز التوجيه الحدودي. يتم وضع جهاز توجيه الحدود في الزاوية بحيث تكون هناك عقدة

Cooja Unit Disk Graph Medium، الذي يقدم الضياع فيما يتعلق بالمسافات النسبية للعقد في وسيط الراديو. كما هو موضح في الجدول التالي، فإن تأخير البدء start delay هو تأخير البدء الأولي للتطبيق لبدء إرسال الرسائل إلى عقدة المصدر. وقت البدء الأولي هذا هو الوقت التقريبي الكافي لتقارب الشبكة الأولي. وهذا يضمن أيضاً عدم فقدان الحزمة المرسلة إلى الخادم بسبب نقص اتصال الشبكة. لذلك يمكن إجراء تقييم صحيح لعدد الحزم المرسلة. قمنا بتعيين RPL mode of operation \downarrow No Downward routes لأننا مهتمون باستخدام multipoint to point traffic لهذا البحث. يتم تعيين DIO Min و DIO Doublings على القيم الافتراضية في ContikiRPL. تمثل نسبة الاستقبال (RX) مدى ضياع الوسيط الراديوي ويتم تعيينه بالنسب المئوية أثناء التكرار المتتالي للمحاكاة. يتم تعيين نسبة الإرسال (TX) على 100% (بدون خسارة) لأننا لا نهدف إلى إحداث خسائر عند طرف الإرسال ولكن فقط عند طرف الاستقبال. تم ضبط نطاق TX على 50 م ونطاق التداخل على 55 م.

الجدول (1) اعدادات المحاكاة.

| Parameters | Value |
|-----------------------|-------------------|
| Start Delay | 65 s |
| RPL MOP | NO_DOWNWARD_ROUTE |
| OF | ETX |
| DIO Min | 12 |
| DIO Doublings | 8 |
| RDC Chanel Check Rate | 16 |
| Send Interval | 4 s |
| RX Ratio | 30-100% |
| TX Ratio | 100% |
| TX Range | 50m |
| Interference Range | 55m |
| Simulation Time | 7 min |
| Client Nodes | 19-39 |

10. التجارب والنتائج: قمنا ومن أجل كل خوارزمية بإعادة التنفيذ عشر مرات وأخذ متوسط القيم الناتجة عن التجارب وذلك بعد استخراج القيم الشاذة والمتطرفة لبعض التجارب بتطبيق قانون المدى الربيعي عن القيم العشر الناتجة.
- مثال لنتيجة تحليل الملف COOJA.testlog باستخدام ملف التحليل analysis.pl الذي قمنا بكتابته بلغة PERL وذلك لخوارزمية Trickle الأصلية:

```

user@instant-contiki:~/contiki-3.0/perl$ perl analysis.pl COOJA.testlog
NETWORK SETUP TIME
=====
First DIO      Last DIO joined DAG      Setup Time(ms)
1136934.000    11603688.000             10466754.000

ENERGY CONSUMPTION
=====
Nodes  Total Transmit ticks  Total Listen ticks  Total Consumption(ticks)  Total cpu  Total lpm  Total Time  %Radio ON Time
20     2816612                2732615              5549227                   7308820   235490414  242799234  2.286

NETWORK TRAFFIC
=====
DIO     DIS     DAO
272     16      71

NETWORK LATENCY
=====
Average Latency(us)  no of SendPackets  Lost Packets
482273              114                 3
    
```

الشكل (4) نتيجة تحليل ملف الخرج COOJA.testlog حالة خوارزمية Trickle وشبكة تحتوي 20 عقدة.

```

user@instant-contiki:~/contiki-3.0/perl$ perl analysis.pl COOJA.testlog
NETWORK SETUP TIME
=====
First DIO      Last DIO joined DAG      Setup Time(ms)
1405564.000    28286209.000             26880645.000

ENERGY CONSUMPTION
=====
Nodes  Total Transmit ticks  Total Listen ticks  Total Consumption(ticks)  Total cpu  Total lpm  Total Time  %Radio ON Time
20     9259804                7457961              16717765                   20139432  478240540  498379972  3.354

NETWORK TRAFFIC
=====
DIO     DIS     DAO
560     37      140

NETWORK LATENCY
=====
Average Latency(us)  no of SendPackets  Lost Packets
1190538              232                 19
    
```

الشكل (5) نتيجة تحليل ملف الخرج COOJA.testlog حالة خوارزمية Trickle وشبكة تحتوي 40 عقدة.

| Setup Time (us) | | | | | | | | | | |
|-----------------|------------|-----------|-------------|-----------|------------|---------------|------------|-----------|------------|--|
| Trickle | Trickle-F | I-trickle | Opt-trickle | E-trickle | Adaptive-k | Trickle-pluse | RD-trickle | Elastic | FL-TRICKLE | |
| 10466754 | 17260865 | 7274411 | 9035959 | 7351746 | 8768046 | 61853419 | 11014344 | 10466754 | 61726613 | |
| 16369081 | 18955983 | 5442737 | | 9194282 | 7808196 | 13679747 | 9841182 | 9533161 | 13679746 | |
| 9533161 | 10660428 | 8788513 | 13258077 | 13916589 | 6542192 | 20662655 | 9917260 | 8782110 | 20662654 | |
| 8782110 | 12763070 | 5948167 | 10208975 | 11039528 | 15819565 | 3197414 | 8983927 | 8502929 | 3197413 | |
| 8498822 | 9681074 | 9733645 | 12492798 | 9083203 | 22056277 | 4733141 | 7733250 | 9686831 | 4733140 | |
| 8502929 | 8128507 | 11899815 | 8749572 | 13923068 | 15943608 | 13672640 | 13076755 | 7375552 | 13672639 | |
| 9086831 | 6314797 | 5348359 | 9562640 | 15018232 | 16042928 | 74792083 | 7046437 | | 61543051 | |
| 7375552 | 7251512 | 8516019 | 9016730 | 9074613 | 11146087 | 14364353 | 10037163 | 9159267 | 14364352 | |
| 14160756 | 13606881 | 11312996 | 15626160 | 11806052 | 11252043 | 4773977 | 11991922 | 10118982 | 4773976 | |
| 9159267 | 5909226 | 5543032 | 6803483 | 6610987 | 9283698 | 4986705 | 12758695 | 4910154 | 4986704 | |
| 10253526.3 | 10253234.3 | 7980769.4 | 10528266 | 10761830 | 12466264 | 21671613.4 | 10240085.5 | 8726193.3 | 20334028.8 | |
| 10.2535 | 10.2532 | 7.980769 | 10.528 | 10.76183 | 12.466 | 21.671 | 10.240 | 8.726 | 20.334 | |

• الحالة الثانية شبكة تحتوي 40 عقدة موزعة بشكل منتظم:

الجدول (3) متوسط زمن التقارب لجميع الخوارزميات حالة شبكة تحوي 40 عقدة.

| Setup Time (us) | | | | | | | | | | |
|-----------------|------------|------------|-------------|------------|------------|---------------|------------|------------|------------|--|
| Trickle | Trickle-F | I-trickle | Opt-trickle | E-trickle | Adaptive-k | Trickle-pluse | RD-trickle | Elastic | FL-TRICKLE | |
| 40855140 | 17416753 | 14648808 | 18690322 | 15697059 | 12159532 | 32735529 | 19773911 | 40855140 | 32238323 | |
| 17349523 | 22843382 | 20303758 | 28176269 | 20042679 | 14385461 | 19246403 | 15118974 | 17349523 | 21370268 | |
| 28357626 | 24867018 | 14207715 | 28031817 | | 23203283 | 21856557 | 16944038 | 28357626 | 61855013 | |
| 59416653 | 29786928 | 21170419 | 11496959 | 15649791 | 18098253 | 61974434 | 13375698 | 59416653 | 39973721 | |
| 17475566 | 16936437 | 115724071 | 22290039 | 17908915 | 18992852 | 21185596 | 22608196 | 17475566 | 31624718 | |
| 26880645 | | 11190896 | 15718811 | 19386832 | 31734759 | 13953223 | 22931517 | 26880645 | 14365945 | |
| 11891187 | 10466855 | 15857995 | 38599504 | 16033311 | 14633906 | 18371649 | 24178340 | 11891187 | 18371648 | |
| 14534196 | 14365159 | 9630225 | 28842332 | 24203226 | 14806070 | 25333581 | 22413630 | 14534196 | 25085222 | |
| 16111137 | 15625516 | 27390313 | 15698690 | 26818759 | 11460205 | 20880805 | 13377894 | 16111137 | 75088078 | |
| 21510616 | 17989376 | 16870917 | 18965362 | 15469556 | 17853004 | 61260884 | 15523752 | 21510616 | 71885237 | |
| 25438228.9 | 19588602.6 | 16699511.7 | 22651010.5 | 17121012.8 | 17732812.5 | 29679866.1 | 18624595 | 25438228.9 | 39177817.3 | |
| 25.438 | 19.58 | 16.69 | 22.65 | 17.12 | 17.73 | 29.67 | 18.624 | 25.438 | 39.177 | |
| 18.62 | | | | | | | | | | |

❖ Traffic Overhead (عدد حزم التحكم):

• الحالة الأولى شبكة تحتوي 20 عقدة موزعة بشكل منتظم:

الجدول (4) متوسط عدد حزم التحكم لجميع الخوارزميات حالة شبكة تحوي 20 عقدة.

| Traffic Overhead | | | | | | | | | | |
|------------------|-----------|-----------|-------------|-----------|------------|---------------|------------|---------|------------|--|
| Trickle | Trickle-F | I-trickle | Opt-trickle | E-trickle | Adaptive-k | Trickle-pluse | RD-trickle | Elastic | FL-TRICKLE | |
| 306 | 300 | 364 | 359 | 388 | 283 | 187 | 364 | 359 | 199 | |
| 332 | 324 | 478 | | 456 | 311 | 171 | 374 | 411 | 177 | |
| 291 | 366 | 358 | 336 | 505 | 285 | 188 | 448 | 377 | 264 | |
| 312 | 314 | 391 | 374 | 421 | 324 | 194 | 354 | 366 | 183 | |
| 288 | 358 | 430 | 410 | 457 | 299 | 184 | 392 | 342 | 196 | |
| 352 | 344 | 372 | 354 | 420 | 329 | 183 | 379 | 382 | 186 | |
| 321 | 295 | 375 | 352 | 429 | 286 | 188 | 521 | | 203 | |
| 382 | 317 | 375 | 364 | 430 | 323 | 184 | 389 | 383 | 181 | |
| 342 | 306 | 418 | 366 | 436 | 323 | 177 | 358 | 353 | 192 | |
| 383 | 333 | 438 | 337 | 512 | 288 | 168 | 441 | 382 | 177 | |
| 330.9 | 325.7 | 399.9 | 361.3 | 445.4 | 305.1 | 182.4 | 402 | 372.7 | 195.8 | |

• الحالة الثانية شبكة تحتوي 40 عقدة موزعة بشكل منتظم:

الجدول (5) متوسط عدد حزم التحكم لجميع الخوارزميات حالة شبكة تحوي 40 عقدة.

| Traffic Overhead | | | | | | | | | | |
|------------------|-----------|-----------|-------------|-----------|------------|---------------|------------|---------|------------|--|
| Trickle | Trickle-F | I-trickle | Opt-trickle | E-trickle | Adaptive-k | Trickle-pluse | RD-trickle | Elastic | FL-TRICKLE | |
| 878 | 856 | 1063 | 782 | 883 | 741 | 409 | 885 | 878 | 436 | |
| 933 | 778 | 916 | 767 | 1106 | 642 | 377 | 763 | 933 | 376 | |
| 755 | 777 | 871 | 766 | | 796 | 366 | 946 | 775 | 389 | |
| 971 | 877 | 915 | 800 | 1086 | 635 | 575 | 831 | 971 | 458 | |
| 720 | 819 | 829 | 832 | 873 | 666 | 438 | 910 | 720 | 447 | |
| 737 | | 787 | 855 | 856 | 697 | 347 | 853 | 737 | 415 | |
| 931 | 932 | 921 | 1058 | 828 | 627 | 359 | 785 | 931 | 406 | |
| 795 | 1035 | 815 | 842 | 843 | 701 | 405 | 823 | 795 | 458 | |
| 806 | 728 | 800 | 870 | 874 | 662 | 360 | 816 | 806 | 487 | |
| 788 | 829 | 803 | 730 | 910 | 624 | 407 | 959 | 788 | 406 | |
| 831.4 | 847.8 | 872 | 830.2 | 917.6 | 679.1 | 404.3 | 857.1 | 833.4 | 427.8 | |

❖ :Network Latency(us)

- الحالة الأولى شبكة تحتوي 20 عقدة موزعة بشكل منتظم:

الجدول (6) متوسط زمن تأخير الحزم لجميع الخوارزميات حالة شبكة تحوي 20 عقدة.

| NETWORK LATENCY(us) | | | | | | | | | | |
|---------------------|-----------|-----------|-------------|-----------|------------|---------------|------------|----------|------------|--|
| Trickle | Trickle-F | I-trickle | Opt-trickle | E-trickle | Adaptive-k | Trickle-pluse | RD-trickle | Elastic | FL-TRICKLE | |
| 532308 | 442046 | 370629 | 489332 | 499127 | 588875 | 575570 | 766419 | 482273 | 502468 | |
| 449327 | 583388 | 534989 | | 626040 | 367911 | 673825 | 521314 | 508353 | 635499 | |
| 493407 | 480281 | 303907 | 402351 | 366522 | 675857 | 1131990 | 427505 | 532000 | 885034 | |
| 518666 | 674179 | 719615 | 680470 | 699431 | 547679 | 778853 | 492048 | 447602 | 801023 | |
| 311874 | 554524 | 449912 | 399541 | 454873 | 568725 | 625802 | 443054 | 245408 | 638368 | |
| 457009 | 431064 | 553427 | 505348 | 501148 | 485115 | 578488 | 416755 | 420112 | 776883 | |
| 492193 | 269005 | 431669 | 360577 | 398143 | 524285 | 700942 | 363738 | | 633860 | |
| 420112 | 557565 | 634134 | 435154 | 611522 | 447170 | 782844 | 310658 | 423946 | 612907 | |
| 372321 | 421862 | 726285 | 548738 | 558521 | 457499 | 672623 | 397087 | 613774 | 625362 | |
| 423946 | 617363 | 352092 | 421967 | 429070 | 549314 | 618879 | 406686 | 683575 | 626840 | |
| 447116.3 | 487127.7 | 507665.9 | 471497.5 | 514439.7 | 521243 | 713981.6 | 454526.4 | 484115.8 | 673824.4 | |
| 0.447 | 0.4871 | 0.5076 | 0.4714 | 0.5144 | 0.5212 | 0.713 | 0.4545 | 0.4841 | 0.6738 | |

- الحالة الثانية شبكة تحتوي 40 عقدة موزعة بشكل منتظم:

الجدول (7) متوسط زمن تأخير الحزم لجميع الخوارزميات حالة شبكة تحوي 40 عقدة.

| NETWORK LATENCY(us) | | | | | | | | | | |
|---------------------|-----------|-----------|-------------|-----------|------------|---------------|------------|-----------|------------|--|
| Trickle | Trickle-F | I-trickle | Opt-trickle | E-trickle | Adaptive-k | Trickle-pluse | RD-trickle | Elastic | FL-TRICKLE | |
| 1454203 | 1080722 | 1161431 | 1135481 | 1460019 | 1034950 | 1186877 | 1179028 | 1454203 | 1063051 | |
| 1582143 | 1198935 | 1141214 | 1083753 | 1031991 | 988334 | 1197598 | 1289462 | 1582143 | 1168330 | |
| 1215329 | 1041179 | 1271024 | 1904790 | | 927241 | 1460989 | 1701357 | 1215329 | 1686731 | |
| 1364689 | 1127835 | 1479384 | 1153663 | 1189765 | 1034345 | 1657518 | 1473176 | 1364689 | 2093450 | |
| 1215359 | 1271652 | 1248620 | 1249815 | 1040516 | 936491 | 1448297 | 1270388 | 1215359 | 1366655 | |
| 1198538 | | 1152059 | 1551966 | 1211083 | 2729894 | 2320783 | 1280337 | 1198538 | 1923897 | |
| 1609353 | 1726095 | 959821 | 2187717 | 1080560 | 1159362 | 1728870 | 1598984 | 1609353 | 1515123 | |
| 1366328 | 1209077 | 1633186 | 1664013 | 1926215 | 1250753 | 2358041 | 1582369 | 1366328 | 1855712 | |
| 953284 | 968184 | 1492163 | 1482670 | 1619380 | 1083372 | 1620371 | 1143305 | 953284 | 1574257 | |
| 881288 | 1431062 | 1139269 | 1169152 | 1206952 | 1313993 | 1809058 | 1354689 | 881288 | 1712824 | |
| 1283251.4 | 1105474.1 | 1267817.1 | 1105474.1 | 1176648.1 | 1245873.5 | 1678840.2 | 1387309.5 | 1283251.4 | 1595003 | |
| 1.283 | 1.105 | 1.267 | 1.105 | 1.176 | 1.245 | 1.678 | 1.387 | 1.283 | 1.595 | |

❖ :Packet delivery Ratio(%)

- الحالة الأولى شبكة تحتوي 20 عقدة موزعة بشكل منتظم:

الجدول (8) متوسط معدل تسليم الحزم لجميع الخوارزميات حالة شبكة تحوي 20 عقدة.

| Packet delivery ratio (PDR) | | | | | | | | | | |
|-----------------------------|-----------|-----------|-------------|-----------|------------|--------------|------------|---------|------------|-------|
| Trickle | Trickle-F | I-trickle | Opt-trickle | E-trickle | Adaptive-k | Trickle-plus | RD-trickle | Elastic | FL-TRICKLE | |
| 0.964 | 0.982 | 1 | 1 | 0.973 | 0.964 | 0.956 | 0.955 | 0.973 | 0.973 | 0.973 |
| 0.991 | 1 | 0.973 | 0.991 | 0.973 | 0.991 | 0.991 | 0.964 | 0.991 | 0.973 | 0.973 |
| 0.991 | 0.991 | 0.991 | 0.991 | 0.973 | 0.982 | 0.938 | 0.991 | 0.973 | 0.929 | 0.929 |
| 1 | 0.964 | 0.956 | 0.964 | 0.991 | 0.991 | 0.964 | 0.991 | 0.991 | 0.956 | 0.956 |
| 0.982 | 0.982 | 0.982 | 1 | 0.973 | 0.982 | 0.982 | 0.991 | 0.991 | 0.982 | 0.982 |
| 0.982 | 0.982 | 0.982 | 0.982 | 0.982 | 0.973 | 0.964 | 1 | 0.991 | 0.947 | 0.947 |
| 0.982 | 0.991 | 1 | 0.964 | 0.982 | 0.956 | 0.964 | 1 | 0.991 | 0.956 | 0.956 |
| 0.991 | 0.973 | 0.964 | 1 | 1 | 0.973 | 1 | 1 | 0.964 | 0.991 | 0.991 |
| 0.947 | 1 | 0.982 | 0.991 | 0.956 | 0.982 | 0.991 | 0.991 | 0.956 | 0.991 | 0.991 |
| 0.964 | 0.991 | 0.991 | 1 | 0.973 | 0.982 | 0.973 | 1 | 0.991 | 0.991 | 0.991 |
| 0.979 | 0.985 | 0.982 | 0.988 | 0.977 | 0.980 | 0.969 | 0.988 | 0.980 | 0.969 | 0.969 |
| 97.9 | 98.5 | 98.2 | 98.8 | 97.7 | 98.0 | 96.9 | 98.8 | 98.0 | 96.9 | 96.9 |

- الحالة الثانية شبكة تحتوي 40 عقدة موزعة بشكل منتظم:

الجدول (9) متوسط معدل تسليم الحزم لجميع الخوارزميات حالة شبكة تحوي 40 عقدة.

| Packet delivery ratio (PDR) | | | | | | | | | | |
|-----------------------------|-----------|-----------|-------------|-----------|------------|--------------|------------|---------|------------|-------|
| Trickle | Trickle-F | I-trickle | Opt-trickle | E-trickle | Adaptive-k | Trickle-plus | RD-trickle | Elastic | FL-TRICKLE | |
| 0.858 | 0.952 | 0.965 | 0.935 | 0.918 | 0.948 | 0.935 | 0.897 | 0.858 | 0.948 | 0.948 |
| 0.883 | 0.926 | 0.913 | 0.931 | 0.948 | 0.970 | 0.940 | 0.918 | 0.883 | 0.939 | 0.939 |
| 0.944 | 0.957 | 0.944 | 0.790 | 0.927 | 0.927 | 0.863 | 0.892 | 0.944 | 0.863 | 0.863 |
| 0.867 | 0.923 | 0.833 | 0.961 | 0.944 | 0.939 | 0.883 | 0.871 | 0.867 | 0.785 | 0.785 |
| 0.940 | 0.948 | 0.944 | 0.944 | 0.952 | 0.957 | 0.888 | 0.905 | 0.940 | 0.914 | 0.914 |
| 0.918 | 0.931 | 0.931 | 0.927 | 0.931 | 0.755 | 0.819 | 0.943 | 0.918 | 0.875 | 0.875 |
| 0.832 | 0.840 | 0.974 | 0.837 | 0.952 | 0.931 | 0.880 | 0.901 | 0.832 | 0.884 | 0.884 |
| 0.880 | 0.901 | 0.849 | 0.846 | 0.841 | 0.948 | 0.816 | 0.841 | 0.880 | 0.824 | 0.824 |
| 0.952 | 0.884 | 0.935 | 0.853 | 0.892 | 0.935 | 0.940 | 0.939 | 0.952 | 0.896 | 0.896 |
| 0.961 | 0.935 | 0.965 | 0.935 | 0.922 | 0.893 | 0.841 | 0.913 | 0.961 | 0.858 | 0.858 |
| 0.903 | 0.918 | 0.925 | 0.895 | 0.922 | 0.920 | 0.88 | 0.902 | 0.903 | 0.878 | 0.878 |
| 90.3 | 91.8 | 92.5 | 89.5 | 92.2 | 92.0 | 88.0 | 90.2 | 90.3 | 87.8 | 87.8 |

❖ **Energy Consumption**: لحساب استهلاك الطاقة نستخدم أداة Powertrace المتوفرة في Contiki التي تقوم بتحديد ملامح الطاقة على مستوى الشبكة للشبكات اللاسلكية منخفضة الطاقة، حيث تقدر استهلاك الطاقة من قبل وحدة المعالجة المركزية، ونقل الحزم والاستماع . يتم ارسال هذه المعلومات إلى الأداة collect view التي تعرضه على الشكل التالي:

تقييم ومقارنة أداء التحسينات على خوارزمية التقطير في انترنت الأشياء

| Node Control | | Sensor Map | | Network Graph | | Sensors | | Network | | Power | | Node Info | | Serial Console | |
|--------------|---------|------------|-------|---------------|----------|---------|-------|-----------------|---------|----------|-----------|--------------|----------------|----------------|-------|
| Node | Rece... | Dups | Lost | Hops | Rtmetric | ETX | Churn | Beacon Interval | Reboots | PU Power | LPM Power | Listen Power | Transmit Power | Power | |
| 1.0 | 6 | 0 | 0 | 0.000 | 0.000 | 0.000 | 0 | 5 min, 05 sec | 0 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 2.0 | 6 | 0 | 0 | 1.000 | 512.000 | 18.000 | 0 | 6 min, 11 sec | 0 | 0.087 | 0.161 | 0.375 | 0.165 | 0.789 | |
| 3.0 | 6 | 0 | 0 | 1.000 | 512.000 | 18.000 | 0 | 5 min, 27 sec | 0 | 0.093 | 0.161 | 0.556 | 0.118 | 0.929 | |
| 4.0 | 6 | 0 | 0 | 1.000 | 512.000 | 16.000 | 0 | 5 min, 27 sec | 0 | 0.166 | 0.158 | 0.758 | 0.251 | 1.334 | |
| 5.0 | 6 | 0 | 0 | 3.000 | 1024.000 | 48.000 | 0 | 5 min, 16 sec | 0 | 0.144 | 0.159 | 0.569 | 0.543 | 1.415 | |
| 6.0 | 6 | 0 | 0 | 2.000 | 768.000 | 32.000 | 0 | 5 min, 27 sec | 0 | 0.098 | 0.161 | 0.583 | 0.307 | 1.148 | |
| 7.0 | 6 | 0 | 0 | 2.000 | 768.000 | 32.000 | 0 | 5 min, 27 sec | 0 | 0.227 | 0.157 | 0.847 | 1.018 | 2.248 | |
| 8.0 | 6 | 0 | 0 | 2.000 | 768.000 | 32.000 | 0 | 5 min, 05 sec | 0 | 0.456 | 0.150 | 1.478 | 2.419 | 4.502 | |
| 9.0 | 6 | 0 | 0 | 4.000 | 1280.000 | 64.000 | 0 | 5 min, 27 sec | 0 | 0.124 | 0.160 | 0.550 | 0.411 | 1.245 | |
| 10.0 | 6 | 0 | 0 | 3.000 | 1024.000 | 48.000 | 0 | 6 min, 11 sec | 0 | 0.292 | 0.155 | 1.164 | 1.351 | 2.962 | |
| 11.0 | 6 | 0 | 0 | 3.000 | 1024.000 | 48.000 | 0 | 5 min, 05 sec | 0 | 0.172 | 0.158 | 0.701 | 0.727 | 1.758 | |
| 12.0 | 6 | 0 | 0 | 3.000 | 1024.000 | 48.000 | 0 | 4 min, 54 sec | 0 | 0.124 | 0.160 | 0.709 | 0.413 | 1.406 | |
| 13.0 | 6 | 0 | 0 | 4.000 | 1280.000 | 64.000 | 0 | 5 min, 16 sec | 0 | 0.123 | 0.160 | 0.522 | 0.388 | 1.193 | |
| 14.0 | 6 | 0 | 0 | 4.000 | 1280.000 | 64.000 | 0 | 5 min, 27 sec | 0 | 0.226 | 0.157 | 0.833 | 0.959 | 2.175 | |
| 15.0 | 6 | 0 | 0 | 5.000 | 1536.000 | 80.000 | 0 | 6 min, 00 sec | 0 | 0.099 | 0.161 | 0.501 | 0.258 | 1.018 | |
| 16.0 | 6 | 0 | 0 | 4.000 | 1280.000 | 64.000 | 0 | 6 min, 11 sec | 0 | 0.122 | 0.160 | 0.525 | 0.385 | 1.191 | |
| 17.0 | 6 | 0 | 0 | 5.000 | 1536.000 | 80.000 | 0 | 5 min, 05 sec | 0 | 0.130 | 0.160 | 0.583 | 0.443 | 1.315 | |
| 18.0 | 6 | 0 | 0 | 4.000 | 1280.000 | 64.000 | 0 | 6 min, 00 sec | 0 | 0.098 | 0.161 | 0.459 | 0.254 | 0.972 | |
| 19.0 | 6 | 0 | 0 | 6.000 | 1792.000 | 96.000 | 0 | 6 min, 00 sec | 0 | 0.109 | 0.160 | 0.489 | 0.383 | 1.140 | |
| 20.0 | 6 | 0 | 0 | 5.000 | 1536.000 | 80.000 | 0 | 5 min, 05 sec | 0 | 0.086 | 0.161 | 0.426 | 0.243 | 0.916 | |
| Avg | 6.000 | 0.000 | 0.000 | 3.263 | 1091.368 | 52.211 | 0.000 | 5 min, 30 sec | 0.000 | 0.157 | 0.159 | 0.665 | 0.581 | 1.561 | |

الشكل (8) استهلاك الطاقة بالميلي واط لجميع العقد في الشبكة التي تحتوي 20 عقدة حالة خوارزمية Trickle الأصلية.

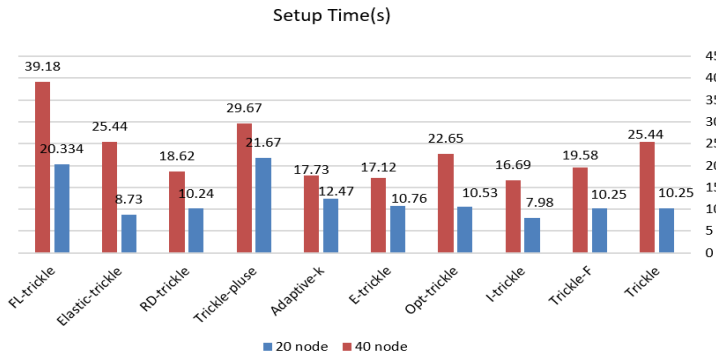
كما في الشكل السابق فإنه ومن أجل كل خوارزمية ومن أجل شبكة بعدد عقد 20 وشبكة بعدد عقد 40 يوجد قيمة للطاقة Power سنزيتها في الجدول التالي :

الجدول (10) متوسط استهلاك الطاقة لجميع الخوارزميات حالة شبكة تحوي 20 عقدة و 40 عقدة.

| | Trickle | Trickle-F | l-trickle | Opt-trickle | E-trickle | Adaptive-k | Trickle-pluse | RD-trickle | Elastic-trickle | FL-trickle |
|---------|---------|-----------|-----------|-------------|-----------|------------|---------------|------------|-----------------|------------|
| 20 node | 1.6 | 1.6 | 2.3 | 1.7 | 1.8 | 1.4 | 1.8 | 1.5 | 1.6 | 1.7 |
| 40 node | 2.3 | 2.3 | 2.2 | 2.3 | 2.2 | 2.1 | 1.9 | 2.6 | 2.3 | 2.5 |

1-10. مناقشة النتائج:

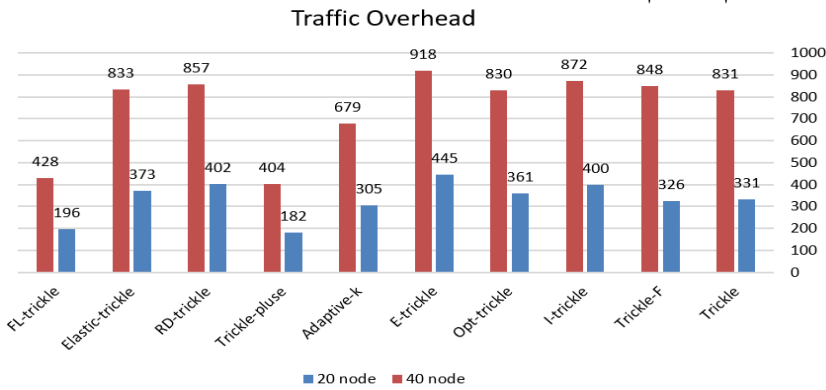
• زمن التقارب:



الشكل (9) مخطط لمقارنة زمن تقارب جميع الخوارزميات.

نلاحظ أنه كلما زاد حجم الشبكة زاد معه زمن التقارب بسبب زيادة عدد العقد التي يجب أن تتضمن إلى المخطط الشجري الخاص بالشبكة. تُبين النتائج أنه كل من Trickle-F و Opt-trickle و E-trickle و RD-trickle لم يظهروا تحسن في زمن التقارب عند عدد العقد المنخفض 20 بينما يبدو تأثيرهم على زمن التقارب عند زيادة حجم الشبكة بمعنى آخر أنه كلما ازداد عدد العقد تخفض هذه الخوارزميات من زمن التقارب. نلاحظ أن خوارزمية Adaptive-k زادت من زمن التقارب عند عدد عقد صغير وقللته عند عدد عقد كبير لذلك سنعتبرها حيادية وليست ذات تأثير على زمن التقارب. بينما نلاحظ أنه كل من Trickle-plus و FL-trickle لا تقدمان أي تحسين على زمن التقارب بل بالعكس يزداد زمن التقارب عند تطبيقهما ضمن بروتوكول RPL، أما Elastic-trickle فتقدم نفس زمن تقارب خوارزمية Trickle الأصلية. الخوارزمية التي تقدم أفضل زمن تقارب هي I-trickle حيث قللت من هذا الزمن عند حجم الشبكة الصغير 20 عقدة والكبير 40 عقدة، وذلك لأنها تأخذ بعين الاعتبار الرسائل التي يتم سماعها من الزمن t وحتى نهاية الفاصل الزمني.

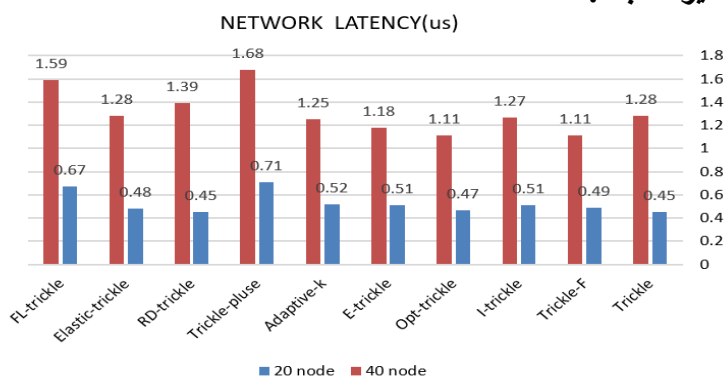
• عدد حزم التحكم:



الشكل (10) مخطط لمقارنة متوسط عدد حزم التحكم لجميع الخوارزميات.

كل من خوارزمية I-trickle و Opt-trickle و E-trickle و RD-trickle و Elastic-trickle تزيد من عدد حزم التحكم المرسل في الشبكة مهما كان حجم الشبكة. تحافظ خوارزمية Trickle-F على عدد الحزم المرسل في الشبكة كما في حالة Trickle الأصلية. كل من Adaptive-k و Trickle-plus و FL-trickle تقلل من عدد حزم التحكم المرسل في الشبكة، وبالتالي تحسن من قيمة هذا المعيار وتعد أفضلها Trickle-plus ، وذلك لأنها تقوم بالاستغناء عن الطول الزمني بين وقت بدء الإزاحة ووقت انتهاءها مما يخفف عدد رسائل التحكم الذي يتناقص كلما زاد طول فاصل الإزاحة الزمني.

• زمن تأخير الشبكة:

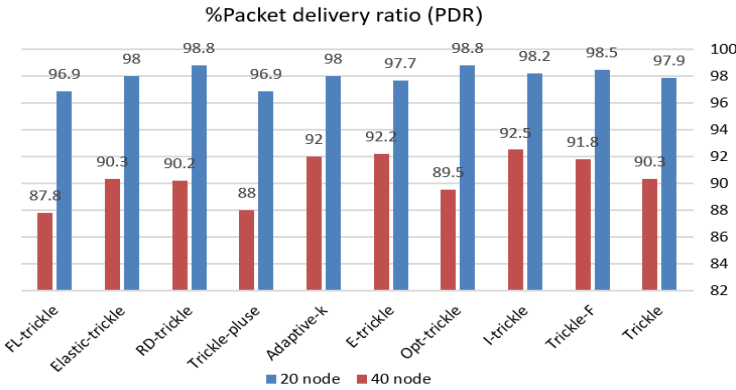


الشكل (11) مخطط لمقارنة متوسط زمن تأخير الحزم لجميع الخوارزميات.

كلما زاد حجم الشبكة زاد متوسط زمن التأخير وذلك بسبب زيادة عدد القفزات التي تسلكها الحزمة حتى تصل إلى المصرف، كما نلاحظ أن تأثير كل خوارزمية على زمن التأخير يبدو بشكل أوضح عند حجم الشبكة الكبير. كل من خوارزمية Trickle-F و Opt-trickle تقلل من زمن تأخير الشبكة، وذلك بسبب جعلها فترة الاستماع فقط أقصر من باقي الخوارزميات وخاصة عند اكتشاف عدم اتساق في الشبكة كما في حالة Opt-trickle. بينما كل من خوارزمية I-trickle و Adaptive-k و RD-

trickle و Trickle-plus و FL-trickle تزيد من زمن تأخير الشبكة. أما Elastic-trickle فتزيد من زمن التأخير. بينما E-trickle فإنها كما نلاحظ تزيد من هذا الزمن عند حجم شبكة صغير و تنقصه عندما يزداد حجم الشبكة لذلك سنعتبرها حيادية بالنسبة لهذا الزمن.

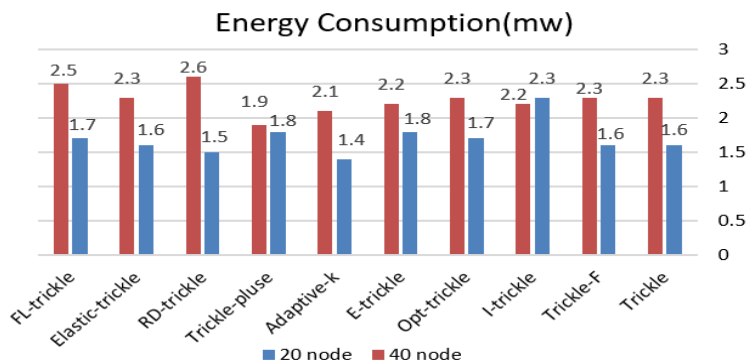
● معدل تسليم الحزم:



الشكل (12) مخطط لمقارنة متوسط معدل تسليم الحزم لجميع الخوارزميات.

نلاحظ من المخطط أنه كلما زاد حجم الشبكة كلما نقص معدل تسليم الحزم ونلاحظ أنّ كل من خوارزمية Trickle-F و E-trickle و Adaptive-k و RD-trickle و Elastic trickle تحسن معدل تسليم الحزم، ونجد أنّ كل من Trickle-plus و FL-trickle تقلل معدل تسليم الحزم بشكل واضح. نلاحظ أنّ I-Trickle تعتبر أكثر خوارزمية تحسن معدل تسليم الحزم.

● استهلاك الطاقة:



الشكل (13) مخطط لمقارنة متوسط استهلاك الطاقة لجميع الخوارزميات.

نلاحظ أنه كلما ازداد حجم الشبكة كلما ازداد استهلاك الطاقة. تخفض Adaptive-k استهلاك الطاقة وذلك بسبب تخفيف مقدار الطاقة الذي تصرفه العقدة على عمليات الإرسال الغير مهمة التي تحصل بسبب قيمة k الغير مناسبة لبنية الشبكة. تزيد Opt-trickle و E-trickle و FL-trickle و Trickle- الاستهلاك الطاقة. ل Elastic-trickle و Trickle-F نفس مستوى استهلاك الطاقة ل Trickle الأصلية.

10-2. ملخص النتائج:

بناءً على ما سبق يمكننا رسم الجدول التالي الذي يحوي ملخص النتائج حيث ترمز الإشارة (+) إلى أن الخوارزمية المقابلة قدمت أداء أفضل من الخوارزمية الأصلية بالنسبة للمعيار المقابل وترمز الإشارة(-) إلى أن الخوارزمية المقابلة قد قدمت أداء أسوأ من الخوارزمية الأصلية بالنسبة للمعيار المقابل بينما ترمز الإشارة (*) إلى أن الخوارزمية المقابلة قدمت أفضل أداء من بين جميع الخوارزميات الباقية والمربع الفارغ يعني أن هذه الخوارزمية لا تؤثر على المعيار المقابل:

الجدول (11) تلخيص النتائج.

| Energy Cosum | PDR | Latency | Traffic Overhead | Setup Time | Metric Algorithm |
|-----------------|-----|---------|---------------------|---------------|---------------------|
| | + | + | | + | Trickle-F |
| - | * | - | - | * | I-trickle |
| - | | * | - | + | Opt-trickle |
| - | + | | - | + | E-trickle |
| * | + | - | + | | Adaptive-k |
| | - | - | * | - | Trickle-plus |
| | + | - | - | + | RD-trickle |
| | + | | - | | Elastic-trickle |
| - | - | - | + | - | FL-trickle |

نلاحظ من الجدول السابق أن الخوارزمية التي تقدم أفضل (أقل) زمن تقارب هي خوارزمية I-trickle ولكن ذلك كان على حساب عدد حزم التحكم المرسل في الشبكة وكمية استهلاك الطاقة اللذين ازدادا عما كانا عليه في Trickle الأصلية. تسلك كل من E-trickle و Opt-trickle نفس سلوك I-trickle حيث تقللان زمن التقارب وبالمقابل تزيدان عدد حزم التحكم واستهلاك الطاقة في الشبكة.

تقدم Trickle-plus أقل عدد حزم تحكم مرسل في الشبكة لكنها بالمقابل تزيد من زمن التقارب وتأخير الحزم وكذلك FL-trickle التي تقلل من عدد حزم التحكم على حساب زيادة زمن التقارب والتأخير وكمية استهلاك الطاقة. تقدم Opt-trickle أقل تأخير للحزم ضمن الشبكة وتقدم I-trickle أفضل زمن تسليم حزم، بينما تقدم Adaptive-k أقل كمية استهلاك طاقة في الشبكة.

3-10. التوصيات:

يمكن استخدام خوارزمية Adaptive-k عندما يكون تخفيض استهلاك الطاقة هو العامل الهام في شبكتنا وذلك لقدرتها على تقليل عدد حزم التحكم ورفع معدل تسليم الحزم. يوصي هذا البحث بإيجاد خوارزمية Trickle تقدم أفضل أداء من ناحية جميع مقاييس الأداء المدروسة في البحث.

11. المراجع:

- [1] Djamaa, B., & Richardson, M. (2015). Optimizing the Trickle Algorithm, 13(9), 10–13.
- [2] Zhao, M., Kumar, A., Han, P., Chong, J., & Lu, R. (2016). A comprehensive study of RPL and P2P–RPL routing protocols : Implementation , challenges and opportunities. *Peer-to-Peer Networking and Applications*, (October 2017).
- [3] Bagula, A., & Pietrosevoli, E. (n.d.). Internet of Things IN 5 DAYS.
- [4] Conference, I., & Hutchison, D. (2013). Internet of Things , Smart Spaces , and Next Generation.
- [5] Qasem, M., Al–dubai, A., Ghaleb, B., Al–dubai, A., Ekonomou, E., Gharibi, W., ... Khalaf, M. B. (2018). A New Load–Balancing Aware Objective Function for RPL ’ s IoT Networks.
- [6] Winter, T., Thubert, P., Corporation, A. R., & Kelsey, R. (n.d.). RFC6550 RPL: IPv6 Routing Protocol for Low–Power and Lossy Networks, 1–157.
- [7] Technology, I. (2017). RPL LOAD BALANCING IN INTERNET OF THINGS, 18(2), 137–150.
- [8] Idrees, A. K., & Witwit, A. J. H. (2018). A Comprehensive Review for {RPL} Routing Protocol in Low Power and Lossy Networks, (September).
- [9] Yassein, M. B., Hmeidi, I., Shehadeh, H., & Yaseen, W. B. (2015). Performance Evaluation of “ Dynamic Double Trickle Timer Algorithm ” in RPL for Internet of Things (IoT).

- [10] P. Levis, N. Patel, D. Culler, and S. Shenker, Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks, in In Proceedings of the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI, 2004, pp. 15–28.
- [11] Vallati, C., & Mingozi, E. (2013). Trickle-F : fair broadcast suppression to improve energy-efficient route formation with the RPL routing protocol.
- [12] Yassein, M. B., Mohammad, R., Masadeh, T., World, T., & Science, I. (2017). A New Dynamic Trickle Algorithm for Low Power and Lossy Networks, (September 2016).
- [13] Ph . D . Dissertation Contributions to the Performance Evaluation and Improvement of the RPL protoocol . (n.d.).
- [14] Ghaleb, B., Al-dubai, A., & Ekonomou, E. (2015). E-Trickle: Enhanced Trickle Algorithm for Low-Power and Lossy Networks, (September).
- [15] Meyfroyt, T. M. M., Stolikj, M., & Lukkien, J. J. (2015). Adaptive Broadcast Suppression for Trickle-Based Protocols, 0–8.
- [16] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, “Cross-level sensor network simulation with cooja,” in Local computer networks, proceedings 2006 31st IEEE conference on, 2006, pp. 641–648.
- [17] Ghaleb, B., Al-dubai, A., Ekonomou, E., Paechter, B., & Qasem, M. (2016). Trickle-Plus : Elastic Trickle Algorithm for Low- Power Networks and Internet of Things, (MEIoT).

- [18] Yassein, M. B., Mohammad, R., Masadeh, T., World, T., & Science, I. (2017). A New Dynamic Trickle Algorithm for Low Power and Lossy Networks.
- [19] Yassein, M. B., Alnadi, A., & Bataineh, A. (2018). RANDOMIZED DYNAMIC TRICKLE TIMER ALGORITHM FOR INTERNET OF THINGS, 187–197.
- [20] Zhang, K., Kim, J., & Cho, G. (2018). An Efficient and Energy-saving Data Dissemination Mechanism for Low-power and Lossy Networks, 7(4).
- [21] Lamaazi, H., Benamar, N., Kahili, N. E. L., & Taleb, T. (2019). FL-Trickle : New Enhancement of Trickle Algorithm for Low Power and Lossy Networks.
- [22] Yassein, M. B., & Aljawarneh, S. (2017). A new elastic trickle timer algorithm for Internet of Things. Journal of Network and Computer Applications, 89(January), 38–47.
- [23] Benson, D. J. (2016). A Performance Study of RPL with Trickle Algorithm Variants.
- [24] Goyal, S. (2017). A Comparative Analysis of the Trickle Algorithms in Routing Protocol for Low Power and Lossy Networks, 6(4), 18–23.

