مجلة جامعة دمشق للعلوم الهندسية
المجلد 39- العدد 4 (2023): 223-242

Damascus university Journal for engineering sciences
Vol 39 No.4 (2023) : 223-242

تضامُن استخدام الذاكرة المخبئية والتوجيه في الشبكات المتمحورة حول المعلومات

أحمد توهان[1]، د.م. محمد إياد الخيّاط[2]

[1] كلية الهندسة المعلوماتية – جامعة دمشق.

[2] دكتور مدرس في كلية الهندسة المعلوماتية – جامعة دمشق.

الملخص

تهدف الشبكات المتمحورة حول المعلومات ICN إلى تحويل الإنترنت الحالي من نموذج يتمحور حول المضيف إلى نموذج يتمحور حول المحتوى. المكون الرئيسي لـ ICN هو سياسة ملء وإفراغ الذاكرة المخبئية (Caching) وخوارزمية التوجيه (Routing) المتبعة لتلبية طلبات المستخدمين من مخازن التخبئة المؤقتة؛ لكن يوجد سلبيات في التخبئة مثل تكرار النسخ، أو توافر البيانات بالذاكرة المخبئية بعُقد قريبة لجهة المستخدم وعدم مراعاة بروتوكول التوجيه لها، وكذلك زيادة الازدحام على عُقد الشبكة في سياسات التخبئة وخوارزميات التوجيه. نقترح Caching and Routing Jointly (CRJ) لدمج عمل سياسة إدارة الذاكرة المخبئية مع خوارزمية التوجيه لتحقيق أهداف مشتركة ترفع كفاءة عمل ICN. كما نقوم بمحاكاة شاملة باستخدام المحاكي Icarus واستخدام طبولوجيا الإنترنت الواقعية؛ تشير نتائج المحاكاة إلى زيادة معدل إصابة الذاكرة المخبئية وتقليل نُسخ البيانات المتماثلة في عُقد متقاربة من بعضها، كذلك تقليل زمن معالجة طلبات المستخدم، وتخفيف الازدحام والحمل على الشبكة.

الكلمات المفتاحية: الشبكات المتمحورة حول المعلومات، سياسات التخبئة، خوارزميات التوجيه.

تاريخ الإيداع:2022/5/22
تاريخ القبول:2022/7/25

حقوق النشر: جامعة دمشق –سورية، يحتفظ المؤلفون بحقوق النشر بموجب الترخيص **CC BY-NC-SA 04**

Caching and Routing Jointly in Information-Centric Networks (ICN)

Ahmad Tohan[1] , Dr. Iyad Alkhayat[2]

[1]Information Technology Faculty, Damascus University**.**
[2]Dr. Information Technology Faculty, Damascus University.

Abstract

Information-Centric Networks (ICN) aim to transform the current Internet from a host-centric model to a content-centric model. The main component of ICN is the policy of caching and the routing algorithm used to respond to user requests from caches. There are downsides to caching such as redundancy of copies, or the availability of data in the cache with nodes near to the end user, and ignoring it from the routing algorithms, as well as Increase congestion on network nodes in caching policies and routing algorithms. We suggest Caching and Routing Jointly (CRJ) to combine the work of the cache management policy with the routing algorithm to achieve common goals that increase the efficiency of the work of the ICN. We perform a comprehensive simulation using Icarus simulator and using real internet topologies; Simulation results: Increased cache hit rate, reduced data replication in nearby nodes, reduced latency to process user requests, and reduced network congestion and load.

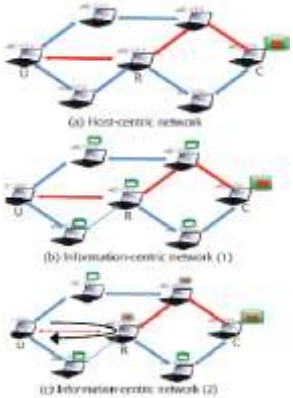
Keywords: Information-Centric Networks, Caching Policies, Routing Algorithms.

Received: 22/5/2022
Accepted: 25/7/2022

Copyright: Damascus University- Syria, The authors retain the copyright under a CC BY- NC-SA

1. المقدمة

في السنوات الأخيرة ظهر نهج جديد من الشبكات في بنية الإنترنت يدعى الشبكات المتمحورة حول المعلومات Information-Centric Networks (ICN)، يقوم على تحويل الاتصال بين الأجهزة في الشبكة من اتصال المضيف للمضيف host-centric Communication model إلى اتصال يتمحور حول المحتوى Content-centric model؛ ميزة مهمة مكنت أجهزة التوجيه من القيام بعملية التخزين المؤقت للمعلومات، ويبقى المحتوى موجود بشكل دائم في مخدمات المحتوى الأصلية (Content custodians) وبالتالي يمكن لأجهزة التوجيه التي تدعم التخبئة، أن تقوم بتخزين المحتوى بشكل مؤقت أثناء مروره [1]. في نموذج الاتصال القائم على المضيف Host-centric كانت تقوم أجهزة التوجيه بتمرير المعلومات من جهاز لآخر حتى الوصول لمخدمات المحتوى الأصلية دون الاهتمام بها أو تخزينها، بينما في النموذج المتمحور حول المعلومات تقوم أجهزة التوجيه بتخزين المعلومات أثناء مرورها، وبالتالي يمكن تقديم ميزتين رئيسيتين: 1- تقديم محتوى إلى مستخدم من جهاز توجيه، حيث يواجه المستخدمون زمن انتقال أقل ومن غير اللازم أن تكون جميع الطلبات موجهة إلى مخدمات المحتوى الأصلية؛ 2- تخفيف الازدحام في الشبكة وبالتالي ليس دوماً الحصول على المعلومات من مسار واحد أي يمكن جلب المعلومات من أجهزة طرفية كما في الشكل (1a)، بفرض أن المستخدم U يطلب معلومات من المخدم C التي تسير على طول المسار باللون الأحمر وذلك وفق خوارزمية التوجيه المتبعة، وعندما يتلقى C الطلب يتم الرد على المستخدم U وفق المسار نفسه. لنقارن السيناريو السابق بطريقة عمل الشبكات المتمحورة حول المعلومات ICN من خلال الشكل (1b) لنفترض أن جميع أجهزة التوجيه في الشبكة قادرة على تخزين المحتوى بشكل مؤقت مع بقاء المحتوى متاح في المخزن C، فعند طلب المستخدم U للمعلومات من المخزن C يقوم بالرد على طلب المستخدم ويتم تخزين المعلومات في المسار باللون الأحمر في كل موجه قد مرت المعلومات عبره. في الشكل (1c) طلب المستخدم U معلومات عندها يستقبل R الطلب ليجد أن المعلومات نُزلت لديه مسبقاً، ليقوم بالرد على طلب U وذلك دون الحاجة إلى إرسال الطلب للمخزن C؛ نلاحظ أنه تم نشر المحتوى من خلال طلب المستخدم، ويمكن أيضاً نشر المحتوى في حالة الرد على طلبات المستخدم وهي موجودة في ذاكرة التخزين المؤقت، أي يمكن إعادة توزيع المحتوى المُخزن بشكل مؤقت دون الاتصال بالمخزن الرئيسي مع الأخذ بعين الاعتبار اهتمامات المستخدم. يقوم تصميم الشبكات المتمحورة حول المعلومات على كيفية تقديم ذاكرة تخزين مؤقتة فعالة مع سياسات التوجيه، للوصول لشبكة تُسرع الوصول للمعلومات وتخفف الازدحام وتقلل النسخ المتكررة من المعلومات وتراعي توزيع الأحمال على العُقد في كافة طبولوجيا الشبكة [2].



الشكل (1) مقارنة بين الشبكات المتمحورة حول المضيف والمتمحورة حول المعلومات

2. مبررات وأهداف البحث

في البداية سنسلط الضوء على أهم الأعمال ذات الصلة بدمج عمل سياسة التخبئة وبروتوكول التوجيه. حيث في العمل [3] تم اقتراح نموذج التحسين المشترك لتوجيه الطلب ومحتوى التخزين المؤقت في الشبكات المتمحورة حول المعلومات، حيث يقوم هذا النهج على خوارزمية بسيطة وقابلة للتكيف مع عُقد الشبكة وتغيرات حركة المرور وخسائر الحزم، وذلك من خلال مفهوم الأوزان الافتراضي لتحديد إمكانيات العُقد في الشبكة، ويكون ذلك بشكل تصاعدي عند نشر المحتوى في الشبكة، كما تعمل الخوارزمية القائمة على الإمكانيات على تحسين النظام بمعدل نقل يصل إلى 18%، وتقلل من تكلفة توجيه المحتوى بنسبة تصل إلى 50%. لكن في هذا العمل تم تقييد دور سياسة التخبئة، أيضاً لم يتم تخصيص عدد محدد من النُسخ في الشبكة مما قد يؤدي إلى تكرار المحتوى، وأيضاً لم يتم بناء الحل بشكل ديناميكي يراعي اهتمامات المستخدمين ضمن فترات زمنية، حيث الخوارزمية تعطي قرار التوجيه والتخبئة اعتماداً على نموذج رياضي ثابت من أجل كل السيناريوهات مع اختلاف الطلبات في طبولوجيا الشبكة. أيضاً في العمل [4] تم اقتراح نموذج للتخزين المؤقت الفعّال واسترجاع المحتوى لتحسين أداء الشبكات المتمحورة حول المعلومات، حيث يقوم هذا النهج على تنسيق كل عُقدة مع مجاوراتها فيما يخص التخزين والاسترجاع، ويتم ذلك بطريقة موزعة للوصول لتخزين مشترك تعاوني للاستفادة من الذاكرة التخزينية للعُقد المتجاورة، وتم تحقيق معدل إصابة للذاكرة المخبئية حوالي 33% من عدد الطلبات الكلي، وتقليل معدل تأخير معالجة الطلبات مقارنةً باستراتيجات ICN الأخرى. لكن في هذا العمل لم يتم تحديد عدد النُسخ الكلي من المحتوى في الشبكة، إنما تم التركيز فقط على معلومات التجاور بين العُقد، كذلك تم تقليل دور بروتوكول التوجيه في بناء الذاكرة المخبئية والاسترجاع منها، وأيضاً المقارنة كانت فقط على نوع واحد من طبولوجيا الشبكات المفترضة في ICN وتم توليدها بشكل عشوائي؛ وبالتالي قد يؤدي ذلك إلى عدم كفاءة عمل الخوارزمية المقترحة في بيئات أخرى، وأيضاً قد لا تناسب الشبكات التي تحوي كميات كبيرة من المعلومات المتبادلة، بسبب تنوع اهتمامات المستخدمين فيها واختلاف وجهات وصولهم. وفي العمل [5] تم اقتراح نموذج للتخزين المؤقت المُشترك مع التوجيه في شبكات الـ ICN، حيث يقوم هذا النهج على تقسيم الشبكة إلى عدة قطاعات جغرافية، ويكون في كل قطاع عُقدة تخزينة (موجه محتوى) تساعد في التوجيه والربط مع القطاعات الأخرى، حيث يتم التوجيه من خلال خوارزمية المسار الأقصر Dijkstra's بالتعاون مع نموذج احتمالي يعتمد على سلاسل ماركوف، وذلك لتسريع الوصول للعُقدة المخبئ ضمنها المحتوى، حيث تم التحسين في معدل إصابة الذاكرة المخبئية وحقق حوالي 24% من إجمالي طلبات التجربة. لكن في هذا العمل لم يتم توزيع الأحمال على كافة عُقد الشبكة، حيث من الممكن أن يحصل ضغط على عُقد قطاع جغرافي دون الآخر، كذلك مع توسع الشبكة قد يصبح من الصعب إدارة العدد الكبير من القطاعات، كذلك لم يتم إدارة تكرار النُسخ بشكل كلي. أما في العمل [6] تم اقتراح نموذج التخزين المؤقت التكيفي والموزع للمحتوى في الشبكات المتمحورة حول المعلومات، حيث يهدف هذا النموذج إلى تخزين أكثر من نُسخة متماثلة للمحتوى في عُقد الشبكة، ليتم تحقيق التوزيع الأمثل ونشر المحتوى بشكل يقلل التأخير للوصول للمحتوى، كما يوفر الاحتفاظ بكائنات المحتوى التي تكون أكثر شيوعاً وشعبيةً. حيث في هذا النموذج تم تحسين

معدل إصابة الذاكرة المخبئية وحقق معدل إصابة حوالي 28%، وكذلك حقق توزيع أحمال على عُقد الشبكة، كذلك تمت إدارة التخزين المخبئي وتقييد عدد النُسخ في الشبكة. لكن في هذا العمل تم تقييد دور بروتوكول التوجيه في المساعدة في التخبئة، أيضاً لم يتم تخصيص نطاق لتكرار النُسخ وربطه بشعبية المحتوى، مما قد يؤدي إلى نشر المحتوى في مكان بعيد عن اهتمامات المستخدمين وقريب من مستخدمين ليس لهم اهتمام به. إذاً نستنتج من الدارسات السابقة أن سياسات التخبئة وبروتوكول التوجيه لا يهتمان في عملهما لكافة الخصائص، حيث تُركز سياسة التخبئة على بعض الخصائص وبروتوكول التوجيه على البعض الآخر وقد يسعون لنفس الهدف مجتمعين، فإن دمج عمل سياسة التخبئة مع بروتوكول التوجيه لتحقيق هدف مشترك، سوف يساعد على: زيادة معدل إصابة الذاكرة المخبئية إلى 37% وتقليل نُسخ البيانات المتماثلة في عُقد متقاربة من بعضها ونشرها بشكل عادل على الشبكة، كذلك تقليل زمن معالجة طلبات المستخدم، وتخفيف الازدحام والحمل في الشبكة.

إن البحث يهدف بشكل أساسي إلى دراسة بروتوكولات التوجيه في ICN مع سياسات التخبئة، للوصول إلى تصميم يحوي على سياسة تخبئة مع بروتوكول توجيه يحقق بُنية ICN متكاملة.

حيث المتطلبات التي يجب توفيرها في سياسة التخبئة (Caching):

- تقليل نُسخ البيانات المتماثلة في عُقد متقاربة من بعضها.
- توزيع الأحمال على عُقد الشبكة لتحقيق إتاحية أكبر.

أما المتطلبات التي يجب توفيرها في بروتوكول التوجيه (Routing):

- تقليل زمن معالجة طلبات المستخدم من خلال جلبها من عُقدة مجاورة، حيث يمكن لبروتوكول التوجيه تتبع مسارها.
- تخفيف الازدحام والحمل في الشبكة.

3. سياسات التخبئة

تتطلب ICN سياسات إدارة الذاكرة المخبئية (التي تحدد المحتوى الذي سيتم تخزينه في الذاكرة المخبئية وما المحتوى الذي سيتم إفراغه)، حيث تقرر تقنيات التخزين المؤقت كيفية معالجة قطع المعلومات التي تمر عبر أجهزة التوجيه، وذلك لاتخاذ القرار بتخزينها وتحديد طريقة استبدالها لاحقاً، ومن هذه السياسات:

3.1- Leave Copy Everywhere (LCE): سياسة تخبئة تقوم على التخزين المؤقت في جميع العُقد (النُسخ في كل مكان من طبولوجيا الشبكة) ضمن مسار الطلب من المصدر حتى المستخدم النهائي، وتكون جميع النُسخ المنتشرة هي متماثلة فيما بينها ضمن جميع العُقد، وتقدم هذه السياسة سرعة وصول للمعلومات لاحقاً وتراعي تنقل العُقد في الشبكة، ولكن تسبب هذه السياسة حمل كبير على الشبكة حيث يتم تكرار النُسخ دون النظر إلى العُقدة المجاورة أو نوعية المعلومات المُخزنة، يؤدي ذلك إلى إغراق الشبكة بمعومات متماثلة متقاربة [7].

3.2- Leave Copy Down (LCD): سياسة تخبئة تقوم على التخزين المؤقت فقط في عُقدة واحدة وتكون مرتبطة مباشرة بمصدر المعلومات (مؤقت أو أساسي)، حيث تقلل النُسخ المتماثلة في الشبكة بشكل كبير، كما تكون النُسخ جميعها أقرب ما يمكن للمستخدم الذي تم تخبئة طلبه، كما تقدم سرعة وصول للمعلومات في حالات محددة فقط مثل طلب معلومات قريبة من المستخدم النهائي (كـ طلبات مكررة في أوقات لاحقة). في هذه السياسة يتم تقليل عدد حالات الوصول السريع للمعلومات

من العُقد التخزينية لقلة عدد النُسخ، حيث يتم تقليل التخزين المكرر في عُقد الشبكة وبالتالي تقليل عدد النُسخ [8].

3.3- Cache Less for More (CL4M): سياسة تخبئة تقوم على التخزين المؤقت فقط في عدد محدد من العُقد المركزية ضمن مسار طلبات المستخدم النهائي، وتقلل النُسخ المتماثلة في الشبكة حيث يتم تحديد العُقدة المركزية من خلال حساب عدد مسارات التوصيل، التي سوف تسير عبرها معلومات المستخدمين من عُقدة ما، وفي حال كان عدد الطلبات التى تسير من العُقدة كثيرة عندها تُحدد كعُقدة تخزين مؤقت؛ التخزين المؤقت في عُقدة أو عدة عُقد مركزية يهدف لتخديم عدد أكبر من طلبات المستخدمين وبالتالي تقليل زمن الحصول على المعلومة، ولكن مع نمو الشبكة قد تؤدي هذه السياسة إلى حدوث حالات اختناق [9].

3.4- Probabilistic Caching: سياسة تخبئة تقوم على التخزين المؤقت اعتماداً على منهج رياضي يحدد معايير تتعلق بالطلب والعُقدة لتحديد إمكانية تخبئة المعلومات، حيث تقوم عُقدة التخبئة بتنفيذ نموذج احتمالي يتعلق بمعلومات الطلب وزمن مروره ومساره من المستخدم حتى المصدر، ومن خلال هذه المعلومات يتم اتخاذ القرار بتخزين المحتوى بشكل مؤقت أم لا، حيث تهدف هذه السياسة لتقليل الاتصال مع العُقد الأخرى لجمع معلومات التخبئة، فهي تعتمد على معلومات تمر عبرها ومن خلالها يتم اتخاذ قرار التخبئة، ولكن هذه السياسة قد لاتعطي دوماً قراراً صحيحاً في التخبئة وخاصة مع تزايد تبادل المعلومات بين عُقد الشبكة واختلاف شعبية المحتوى [10].

3.5- Latency-Aware Caching (LAC): سياسة تخبئة تقوم على التخزين المؤقت اعتماداً على زمن معالجة الطلب لتحديد إمكانية تخبئة معلومات الطلب، فعندما يكون زمن معالجة الطلب استغرق وقتاً طويلاً يتم اتخاذ قرار التخزين المؤقت للمحتوى المنقول عبر الطلب، وذلك من أجل تخديم الطلبات اللاحقة بسرعة أكبر، وتقليل الزمن في الحالة الأولى، والوصول لشبكة نقل معلومات بتأخير زمني أقل [11].

3.6- Popularity Cache: سياسة تخبئة تقوم على التخزين المؤقت للمعلومات الأكثر استخداماً في العُقد الداخلية للشبكة، أما الأقل في العُقد الطرفيّة للشبكة، حيث يتم تحديد المعلومات بأنها شعبية من خلال عدد مرات تكرار طلب هذه المعلومات وتخزن في العُقد المركزية الداخلية للشبكة، أما المعلومات عكس ذلك يتم وضعها في طرفيّات الشبكة للاستفادة منها في تلبية طلبات أخرى في أوقات لاحقة [12].

3.7- Congestion-Aware Caching and Search (CACS): سياسة تخبئة تقوم على التخزين المؤقت للمعلومات التي تسبب ازدحاماً عند طلبها من المستخدمين ليتم تقليلهُ في طلبات أخرى، حيث يتم دوماً حساب الإنتاجية (Throughput) للشبكة ومن خلالها يتم تحديد هل معلومات الطلبات بحاجة للتخزين المؤقت أم لا، فعندما تكون الإنتاجية مرتفعة أي يوجد كمية كبيرة يتم نقلها ومن أجل تخفيف الازدحام في تلبية طلبات لاحقة يتم اتخاذ قرار التخزين المؤقت [13].

3.8- Randomly Copy One (RCO): سياسة تخبئة تقوم على التخزين المؤقت في عُقدة عشوائية ضمن مسار الطلب من المصدر حتى المستخدم النهائي، حيث يتم اختيار عُقدة واحدة في مسار الطلب من أجل التخبئة عليها، ويمكن استخدام هذه السياسة في الشبكات المتمحورة حول المعلومات من أجل تخفيف زمن تحديد العُقد المناسبة للتخبئة ومن أجل نشر المحتوى وتلبية الطلبات وفق خوارزميات عشوائية [14].

4. سياسات الاستبدال

تتطرق سياسات إدارة التخزين المؤقت لكيفية تبديل وإفراغ المحتوى المُخزن مسبقاً في عُقد الشبكة من خلال سياسات التخبئة السابقة، وتساهم سياسات الاستبدال في تجديد المحتوى من أجل حصول المستخدم النهائي على أحدث معلومات كما هي في المصدر الأساسي المُخدم للشبكة [2] وهي:

4.1-FIFO: تقوم هذه السياسة باستبدال المحتوى على مبدأ الرتل (First In First Out) من خلال تحرير الأقدام واستبداله بالأحدث عند إمتلاء الذاكرة المخبئية للعُقدة.

4.2-RANDOM: تقوم هذه السياسة باستبدال المحتوى من خلال استخدام خوارزمية عشوائية، فعند وصول المعلومات إلى العُقدة يتم استخدام تابع اختيار عشوائي في تحرير معلومات مخزنة واستبدالها بمعلومات جديدة.

4.3-LRU: (Least Recently Used) يتم استبدال المحتوى من خلال حذف الأقل استخداماً ضمن فترة زمنية، فيُحدد المحتوى الأكثر شعبية من الأقل منه، ثم يتم الاستبدال بالأقل شعبية عند إمتلاء الذاكرة المخبئية للعُقدة.

4.4-LFU: (Least Frequently Used) يتم استبدال المحتوى من خلال حذف الأقل تكراراً، فيحدد المحتوى الأكثر تداولاً من الأقل منه، ثم يتم الاستبدال بالأقل تداولاً عند إمتلاء الذاكرة.

4.5-TTL: تقوم هذه السياسة (Time To Live) باستبدال المحتوى من خلال تحديد صلاحية زمنية لكل محتوى مخزن داخل الذاكرة المخبئية للعُقدة، وعند انتهاء مدة الصلاحية الزمنية يتم استبداله بمحتوى جديد.

5. بروتوكولات التوجيه

تهتم ICN بخوارزميات توجيه الطلبات ومعالجتها لتلبية حاجات المستخدمين، وذلك من خلال تحديد وجهة البيانات المقابلة للطلب والحصول عليها من خلال مخازن البيانات المخبئة (Caching) في عُقد الشبكة ضمن مسار الطلب، أو جلبها من المصدر الرئيسي ومن ثم تخزينها ضمن العُقد في مسار العودة وأخيراً تسليمها للمستخدم، ويوجد العديد من خوارزميات التوجية المطبّقة في ICN، والتي تعتمد على جداول الذاكرة المخبئية ليتم تحديد مسار البيانات المؤقتة المنتشرة في عُقد الشبكة [15] ومن هذه البروتوكولات:

5.1-Breadcrumbs: بروتوكول توجيه، يقوم في عمله على تخزين مسارات العُقد التي تحوي على المعلومات المخبئة، ويحفظها في جداول تحوي معلومات الطلب ومسار العُقدة الوجهة المخزن عليها المعلومات، وكل مدخل في جدول التوجيه لهذا البروتوكول يدعى Breadcrumb [16].

5.2-Hash-Routing: بروتوكول توجيه يعتمد على تابع تقطيع (Hash-Function)، حيث يقوم التابع بتحديد العُقدة التي تحوي الطلب، عندما يقوم المستخدم بطلب معلومات يتم استدعاء تابع التقطيع لتحديد العُقدة التخزينة وفي حال عدم وجودها يتم الاتصال بالمصدر الأساسي للبيانات. كما يفرض مخطط عمل بروتوكول Hash-Routing وجود عُقد حواف Edge-domain routers وعُقد ذاكرة مخبئية Cache nodes، وبعد ذلك يتم تعريف تابع تقطيع Hash function لربط معرفات المحتوى في عُقد الذاكرة المخبئية، حيث تابع التقطيع يتم استخدامه من قبل عُقد الذاكرة المخبئية، لمعرفة أي محتوى يمكن تخبئته في الموجهات الحواف Edge routers كما يساعد في التوجيه لعُقد الذاكرة لجلب المحتوى المُخزن ضمنها [17].

5.3- Characteristic Time Routing (CTR) : يتم التوجيه في هذا البروتوكول المقترح اعتماداً على اهتمامات

المستخدم ضمن فترة زمنية (Time-to-Live (TTL، حيث يُخبئ المحتوى مع تسجيل مساره في عُقد الشبكة المناسبة، ويحوي جدول التوجيه في CTR على معلومات تتعلق بالمحتوى والذاكرة المخبئية والوقت الذي سيتم بعده حذف سجل التخزين، حيث يُحدد الوقت من خلال اهتمامات المستخدمين ومن الممكن أن يبقى مّخزن لمدة طويلة في الذاكرة المخبئية، ويتم ملء جدول التوجيه بشكل دوري من خلال اهتمامات المستخدمين [18].

5.4- Cache-Aware Routing (CAR): يتم التوجيه اعتماداً على خوارزميات برمجة ديناميكية تحدد معايير النشر والاشتراك وتخزين وتوجيه الطلبات في عُقد الشبكة، تقوم الخوارزميات بحساب معايير خاصة لمسار التوجيه مثل الكلفة ونوعية المحتوى وشعبيته. يفترض CAR أن بنية الشبكة مكونة من الناشرين Publishers والمشتركين Subscribers وموجهات المحتوى CR، كما يكون لكل محتوى لدى الناشر معرف خاص به Content ID ليكون المميز لكائن المحتوى وذلك عند نشره في الشبكة [19].

5.5- Cache Aware Target idenTification (CATT): بروتوكول توجيه مُصمم بشكل أساسي على مكونين وهما التوجيه وتخبئة المحتوى؛ من أجل التوجيه يتم الاعتماد على Potential Based Routing (PBR) حيث يدعم CATT آليات التخزين المخبئي الانتقائي داخلياً أو خارجياً، حيث يحدد نظام التخزين المخبئي الداخلي معايير تخزين ملف المحتوى بشكل مؤقت وفقاً لسياسته الداخلية، وتحدد العملية الخارجية نقطة التخزين المخبئي لتحسين المستوى العام والأداء، وذلك عن طريق توزيع ملفات المحتوى بكفاءة التخزين المخبئي من خلال نقاط عبر الشبكة الواسعة، [14].

6. الخصائص الأساسية لسياسات التخبئة وبروتوكولات التوجيه

يلعب عمل سياسة التخبئة وبروتوكول التوجيه دوراً هاماً في نشر معلومات الشبكة وتوصيل المعلومات بين عُقدها، وبسبب القيود التي تفرضها بنية الشبكات المتمحورة حول المعلومات تم تقسيم عمل السياسة والبروتوكول إلى عدة خصائص أساسية، حيث تعتبر الخصائص التي تقدمها السياسة والبروتوكول معاً هي بمثابة معيار هام لتقييم كفاءة وعمل الشبكة في ICN ورفع من سويّة عملها؛ إن ما تركز عليه معظم الأبحاث بالنسبة لسياسات التخبئة وبروتوكولات التوجيه هو التحسين على عمل السياسة والبروتوكول من خلال تحقيق أكبر عدد من الخصائص والميزات، وأهمها:

6.1- التخبئة ضمن المسار: تُعتبر عملية التخبئة ضمن مسار الطلب on-path Caching من أهم الخصائص التي تقوم بها سياسة التخبئة، وفي الدراسات الحديثة يمكن أن يساعد على القيام بها بروتوكول التوجيه أيضاً، كما تطرقنا سابقاً إلى أن معمارية الشبكات المتمحورة حول المعلومات تفرض نشر المحتوى في عُقد الشبكة بشكل مؤقت، حيث يقوم في البداية المستخدم النهائي بطلب محتوى معين والذي يكون متموضع على مخدم بعيد يستقبل الطلب الموجه المتصل بالمستخدم مباشرة، ومن ثم تقوم الشبكة بتبادل الطلب حتى الوصول للمخدم البعيد ويعاد محتوى الطلب في بعض الحالات بنفس المسار، حيث كل عُقدة في الشبكة (الموجهات CRs Content Routers) تحتفظ لديها بنُسخة من المحتوى ضمن جدول مهيكل حسب السياسة والبروتوكول المستخدم في الشبكة، كما من الممكن أن يكون عدة أساليب للتخبئة وكذلك هيكلة وبناء الذاكرة المخبئية، وفي النهاية يرسل المستخدمون الآخرون طلباتهم واهتماماتهم لعُقد الشبكة لتتلقى هذه العُقد الطلبات وتقوم بالاستعلام على الذاكرة المخبئية لديها، حيث يربط كل محتوى مع معرف خاص

به Content_ID، وفي حال وجد المحتوى يعاد إرساله للمستخدم دون الحاجة لجلب معلومات الطلب من المخدم البعيد [20].

6.2-التخبئة خارج المسار: تعتبر عملية التخبئة خارج مسار الطلب off-path Caching هي أيضاً من أهم الخصائص التي تقوم بها بروتوكولات التوجيه، وذلك من خلال تحديد الموجهات التي ستقوم بعملية التخبئة، حيث يوجد خوارزمية عمل مركزية تتبع في أغلب الأحيان لبروتوكول التوجيه، لتمنع التكرار وتوزع المحتوى بشكل عادل ومناسب جغرافياً لكافة عُقد الشبكة، وعندما يرسل المستخدمون الأخرون طلباتهم واهتماماتهم لعُقد الشبكة، يتم تلقي هذه الطلبات من العُقد، حيث تقوم بالاستعلام على الذاكرة المخبئية لديها في حال وجد المحتوى يتم إرساله للمستخدم دون الحاجة لجلب المعلومات من المخدم الأصلي للمعلومات [20].

6.3-البحث في الجوار: تعتبر عملية البحث في الجوار Neighbor Search هي أيضا خاصية هامة من الخصائص التي تقوم بها بروتوكولات التوجيه، وفي الدراسات الحديثة يمكن أن تساعد على القيام بها سياسة التخبئة أيضاً، كما تطرقنا سابقاً إلى أن معمارية الشبكات المتمحورة حول المعلومات تفرض نشر المحتوى في عُقد الشبكة بشكل مؤقت، ومن ثم البحث عنه بغية تلبية طلبات المستخدمين بأقل تأخير زمني من العُقدة الأقرب جغرافياً؛ في البداية يرسل المستخدم النهائي طلب للحصول على المحتوى، فتستقبل العُقدة المتصلة به مباشرة وتقوم بالبحث في الذاكرة المخبئية لديها، وفي حال وجود المحتوى لدى العُقدة يتم إعادة توجيهه مباشرة للمستخدم، أما في حال عدم وجوده وكانت خاصية البحث في الجوار فعالة يتم إعادة توجيه الطلب لكافة العُقد المتصلة مباشرة بالعُقدة Broadcasts، والعُقدة التي يكون المحتوى مخزن لديها يتم مباشرة إرجاع المحتوى للعُقدة المرسلة للطلب، ومن ثم توصيله للمستخدم النهائي، في هذه الحالة يمكن لسياسة التخبئة المستخدمة في الشبكة من الاستفادة من خاصية البحث في الجوار، وتخزين المعلومات التي لم تكن لديها في الذاكرة المخبئية لها، وبالتالي تقليل زمن الحصول عليها في حال طلبها في فترات زمنية لاحقة من المستخدمين، كما تساعد خاصية البحث في الجوار على نشر المحتوى في العُقد التي لم يكن لديها المحتوى مخزن، وذلك من خلال جلبه من العُقد التي تملك المحتوى وتكون مجاورة لها دون الحاجة للاتصال بالمخدم الأساسي الذي يحوي على المعلومات الأصلية، ويمكن في بعض الحالات تحديد مجال بحث دون البحث في كافة العُقد ويتبع ذلك لسياسات التخبئة وبروتوكولات التوجيه المستخدمة في شبكة ICN [7].

6.4-تحديد عدد النُسخ: تعتبر عملية تحديد عدد النُسخ Limited number of copies هي أيضاً خاصية هامة من الخصائص التي تقوم بها بروتوكولات التوجيه وسياسات التخبئة لأنها توفر المساحة الكافية من حجم الذاكرة المخبئية، وذلك من أجل تخزين كمية كبيرة من المحتوى الغير المتماثل، وبالتالي تقليل عدد مرات الطلب من المخدم البعيد الذي يحوي المعلومات، وذلك لرفع كفاءة عمل الشبكة في ICN، حيث نفترض دوماً في شبكة ICN وجود عدد كبير من الموجهات داخل الشبكة تقوم سياسة التخبئة المستخدمة في الشبكة من تخزين المحتوى، وذلك عند ورود الطلبات عبر الموجهات من المستخدمين النهائيين، يتم التخزين بإحدى الطريقتين أما تخزين ضمن المسار on-path Caching أو تخزين خارج المسار off-path Caching، بعدها يتم تلبية طلبات المستخدمين من ذاكرة التخبئة في الموجهات؛ كما تهدف خاصية تقليل عدد النُسخ إلى عدم التخزين في عُقد متقاربة وبالتالي يقع

على عاتق كل من سياسة التخبئة وبروتوكول التوجيه إدارة النُسخ وتحديد عددها وتنسيق إعادتها للمستخدمين واستبدالها لاحقاً [21] .

7. تضامُن استخدام الذاكرة المخبئية والتوجيه

لنفترض أن شبكة مكونة من مجموعة من العُقد N وكل منها قادر على التخزين المخبئي، وعدد M من مقدمي المحتوى الأصلي للمعلومات (حيث يكون المحتوى متاحاً في أي لحظة)، حيث يكون لكل عُقدة تخزينية في الشبكة ذاكرة مخبئية بحجم C، ويكون حجم الملف K الذي يُنشئ المستخدمون اهتماماتهم له بواسطة عملية بواسون بمعدل λ، وتتبع شعبية المحتوى لتوزيع Zipfian [22]. كما تعتمد الشبكة المفترضة استراتيجية توجيه أصلية (من خلال خوارزمية Dijkstra's التي تعتمد على المسار الأقصر Shortest path routing) أو من خلال بروتوكول توجيه في ICN مثل Hash-Routing؛ كما تقوم العُقد بعملية التخبئة للمحتوى من خلال سياسات التخبئة في ICN مثل LCE أو LCD مع استخدام سياسات الاستبدال مثل LRU، وبالتالي الوصول لشبكة متمحورة حول المعلومات تحوي على عُقد تخزين مخبئي مؤقت، ثم يُنشئ المستخدمون اهتماماتهم بالمحتوى ليتم ملء الذاكرة المخبئية، بعدها يأتي دور بروتوكول التوجيه من أجل تلبية الطلبات من أقرب عُقدة تملك المحتوى (وذلك من أجل تقليل التأخير Latency وزيادة الإنتاجية Throughput)، نتيجة ذلك فإن الخصائص الأساسية لسياسات التخبئة وبروتوكول التوجيه هي من تحدد على أي عُقد سوف تتموضع عليها المعلومات، ومن ثم كيفية جلبها للمستخدمين عند توليد اهتماماتهم.

7.1- الحل المقترح: نقترح Caching and Routing Jointly (CRJ) لدمج عمل سياسة إدارة الذاكرة المخبئية مع خوارزمية التوجيه لتحقيق أهداف مشتركة ترفع كفاءة عمل ICN. حيث نسعى لتحقيق زيادة معدل إصابة الذاكرة المخبئية وتقليل نُسخ البيانات المتماثلة في عُقد متقاربة من بعضها، كذلك تقليل زمن معالجة طلبات المستخدم، وتخفيف الازدحام بين عُقد الشبكة. كما يحقق الحل المقترح بعض الخصائص الأساسية لشبكة ICN، والتي هي معيار هام في تقييم عمل كل سياسة التخبئة وبروتوكول التوجيه، ومن هذه الخصائص التي يحققها الحل المقترح هي: التخبئة على المسار، والتخبئة خارج المسار، والبحث في الجوار، وتحديد عدد النُسخ.

الجدول(1) مقارنة الحل المقترح مع سياسات التخبئة وبروتوكولات التوجيه[18,2]

Name	Category	on-path Caching	off-path Caching	Neighbor Search	Limited number of copies
LCE	Caching	✓	✗	✓	✗
LCD	Caching	✓	✗	✓	✗
CL4M	Caching	✓	✗	✓	✓
ProbCache	Caching	✓	✗	✓	✗
LCC	Caching	✓	✗	✓	✗
PopCache	Caching	✓	✗	✗	✗
CACS	Caching	✓	✗	✓	✗
ECO	Caching	✓	✗	✗	✓
Breadcrumbs	Routing	✓	✓	✗	✗
Hash-routing	Routing	✗	✓	✗	✗
CTR	Routing	✓	✓	✓	✗
CAR	Routing	✓	✗	✓	✗
CATT	Routing	✓	✓	✓	✗
CRJ	Caching Routing	✓	✓	✓	✓

7.2-خوارزمية الحل المقترح CRJ: تعتمد خوارزمية الحل المقترح على بنية المعطيات البيان *Graph*، والذي يتكون من عدة عُقد V (تمثل موجهات الشبكة، والمستخدمون النهائيون، ومخازن البيانات الأصلية)، ووصلات E تمثل مسارات ربط الشبكة التي تصل العُقد فيما بينها؛ كما يجب تحديد عدد النسخ المخبئة Number of Copies ومعلومات طلب المستخدم والمحتوى المهتم به؛ تتم تخبئة البيانات في كل عُقدة من عُقد الشبكة من نوع موجه ويكون لها جدول تخزين مؤقت للبيانات يحوي: معلومات العُقدة، وسياسة الاستبدال، ومعرف المحتوى، وتاريخ ووقت التخزين؛ ويكون الهدف الرئيسي للخوارزمية توصيل المحتوى

وتخبئته للاستفادة في تلبية طلبات أخرى. تُنفذ الخوارزمية مع كل طلب ويتم مع كل طلب تحديد مواقع البيانات Content Locations وعدد النسخ المخبئة Number of Copies، ويتم إيجاد أقرب نسخة للبيانات Nearest Replica من خلال المسار الأقصر حسب خوارزمية Dijkstra (المسار الأقصر هو الأقل تأخيراً)، ويتم تشكيل المسار الرئيسي Primary Path وهو المسار الأقصر من أقرب نسخة حتى المستقبل، وتحديد العُقد الغير ممتلئة بالبيانات Unfilled Cache Nodes (عُقد تكون ذاكرتها المخبئية ذات ضغط تخزين قليل)، ومن الممكن أن يكون للمتحوى أكثر من مصدر أساسي ونحن نركز في دراستنا على وجود مصدر أساسي Custodian واحد فقط. في البداية يرسل المستخدم طلب للحصول على محتوى ما، حيث يستقبل الطلب الموجه المرتبط به مباشرةً ويقوم بمعالجة الطلب، حيث هنا تُنفذ خوارزمية الحل المقترح CRJ كما هو موضح في الجدول (2) ويتم ذلك من خلال توجيه الطلب باتجاه أقرب نسخة Nearest Replica (سواء كانت في المصدر الأساسي أو كـ نسخة مخبئة)؛ ومن ثم يتم تحديد Primary Node وهي العُقدة الأولى من المسار الأقصر Primary Path من جهة النُسخة الأقرب وتملك ذاكرة مخبئية Cache، ومن هذه العُقدة نستخرج كل مجاوراتها والتي تملك ذاكرة مخبئية، ومن هذه المجاورات نحصل على عُقدة التخبئةCache Node وهي العُقدة الأقرب للمستخدم من بين هذه المجاورات.

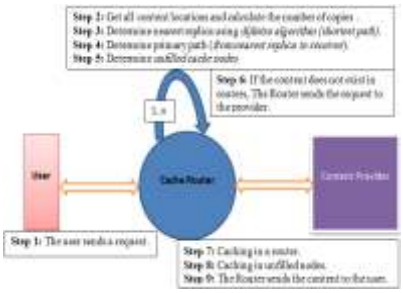
الجدول (2) خوارزمية الحل المقترح CRJ

Algorithm: Caching and Routing Jointly (CRJ).
Input: Graph G(*V, E*) , Request (*receiver, content,max number of copies*).
Output: Cache content and forward content to receiver.

Step 1: Get all content locations and calculate the number of copies .
Step 2: Determine nearest replica using dijkstra algorithm (shortest path).
Step 3: Determine primary path (from nearest replica to receiver).
Step 4: Determine unfilled cache nodes.

```
1.  forward request (from receiver to nearest_replica)
2.  for node in primary_path do
3.  if  node has cache then  primary_node =node ; break  end if
4.   primary_neighbors = all nodes next to the primary_node  have cache
5.  cache_node=the nearest node to receiver in primary_neighbors
6.   if  cache_node != None  then
7.  if  number_of_copies  <  max_number_of_copies  & cache_node in primary_path  then
8.  forward request (from nearest_replica to cache_node)
9.  put content in cache_node
10. nearest_replica=cache_node
11. number_of_copies= number_of_copies  + 1
12. end if
13. if  number_of_copies < max_number_of_copies & cache_node not in primary_path  then
14. cache_node=is selected from cache nodes and is determined by a specified steps
15. forward request (from nearest_replica to cache_node):async
16. put content in cache_node
17. number_of_copies= number_of_copies  + 1
18. end if
19. new_cache_node=the nearest node to receiver in unfilled cache nodes
```

```
20.   new_ nearest_replica=the nearest node to new_cache_node  in unfilled cache nodes
21.  if  number_of_copies < max_number_of_copies & cache_node != new_cache_node then
22.  forward request (from new_ nearest_replica to new_cache_node): async
23.  put content in new_cache_node
24.  end if
25.  end if
26.  forward request (from nearest_replica to receiver)
```

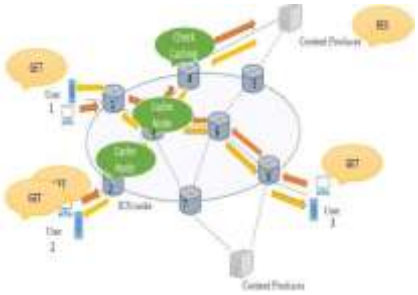


الشكل (2) مخطط عمل خوارزمية CRJ

في المرحلة الأولى [السطر 7] وعند وجود عُقدة التخبئة Cache Node وفي حال لم يتم يتجاوز عدد النُسخ الأعظمي ووجود الـ Cache Node ضمن المسار الأقصر Primary Path، نوجه المحتوى باتجاه الـ Cache Node لتخبئته ضمنها وتصبح النُسخة الأقرب هي الـ Cache Node وزيادة عدد النُسخ نُسخة واحدة؛ المرحلة الثانية [السطر 13] وعند وجود عُقدة التخبئة Cache Node وفي حال لم يتم يتجاوز عدد النُسخ الأعظمي ووجود الـ Cache Node خارج المسار الأقصر Primary Path، نوجه المحتوى باتجاه الـ Cache Node لتخبئته ضمنها ويكون ذلك بشكل غير متزامن وزيادة عدد النُسخ نُسخة واحدة؛ المرحلة الثالثة [السطر 19] نبحث عن عُقدة تخبئة أخرى New Cache Node في العُقد الغير ممتلئة والقريبة من المستخدم النهائي وإيجاد أقرب نُسخة لها New Nearest Replica، وعند وجود عُقدة التخبئة Cache Node وفي حال لم يتم يتجاوز عدد النُسخ الأعظمي و الـ Cache Node ليست هي New Cache Node، نوجه المحتوى من New Nearest Replica باتجاه الـ New Cache Node لتخبئته ضمنها ويكون ذلك بشكل غير متزامن؛ في النهاية يتم توجيه المحتوى من أقرب نُسخة (سواء كانت مصدر أساسي أو نُسخة تخبئة تم توليدها أو مخبئة مسبقاً) للمستخدم النهائي. وبذلك يكون تم تنفيذ التخبئة والتوجيه معاً في خوارزمية الحل المقترح CRJ؛ التوجيه من خلال النُسخة الأقرب والتخبئة إما في ضمن مسار الطلب أو خارج مسار الطلب كما هو موضح في الشكل(2). حيث نعتمد التخبئة ضمن المسار في حال وجدت عُقدة التخزين ضمن مسار الطلب، ونعتمد التخبئة خارج المسار في حال وجدت عُقدة التخزين خارج مسار الطلب أو كانت من العُقد الغير ممتلئة والقريبة من المستخدم النهائي. أما عند امتلاء الذاكرة المخبئية لجميع العُقد فنلجأ للاستبدال وتم استخدام سياسة الاستبدال FIFO (First In First Out) حيث تتعامل هذه السياسة في استبدال المحتوى على مبدأ الرتل، من خلال تحرير المحتوى المُخزن أولاً واستبداله بالمحتوى الجديد.

تفترض خوارزمية CRJ التخبئة أقرب مايمكن للمستخدم وذلك عند اهتمامه بـ بيانات متماثلة في لحظات زمنية متقاربة؛ في البداية يُحسب المسار الأقصر بين المصدر الأساسي والمستخدم النهائي وهو الأقل تأخيراً، ويتم اختيار

عُقدة التخبئة من جهة المصدر الأساسي من عُقد المسار الأقصر، فإذا طلب البيانات ذاتها يتم تقريبها خطوة حتى تصبح أقرب ما يمكن.

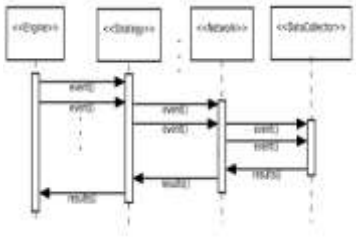


الشكل (3) مثال توضيحي لعمل خوارزمية CRJ

في الشكل(3) مثال توضيحي لعمل خوارزمية CRJ، في البداية يقوم المستخدم User1 بإرسال طلب GET للحصول على محتوى وليكن Content1، ليستقبل الموجه (1) المرتبط به الطلب ويقوم بتنفيذ خطوات الخوارزمية، حيث يقوم أولاً بالاستعلام Q من جميع الموجهات المتصلة به وبمسافات قريبة منهُ، ويكون الرد أما بوجود المحتوى ضمنها أم لا كما يكون الرد أيضاً معلومات عن الموجهات الغير ممتلئة أو ذات الانشغالية الأقل، ومن ثم يتم حساب عدد النُسخ لهذا المحتوى، وحساب المسار الأقصر باتجاه المحتوى (قد يكون مسار باتجاه موجه مخبئي أو مسار باتجاه المخزن الأصلي في حال عدم وجوده في عقد التخزين المخبئي)، حيث سنفترض أن المعلومات غير موجودة بشكل مؤقت وسيتم جلبها من المخزن الأصلي Content Producer فيرسل الموجه(1) الطلب باتجاهه، ليقوم بالرد بالمحتوى RES ليتلقى المحتوى الموجه (4) ويقوم أيضاً بتحديد أقرب موجه باتجاه المستخدم User1 وهو الموجه (3) ويُحدد كـ موجه تخزين مخبئي Cache Node وتتم عملية التخزين المؤقت ضمنه، ويرسل الموجه(3) المحتوى باتجاه الموجه (1) كما يقوم بتخزينه أيضاً في موجهات أخرى حسب الانشغالية ومن ثم إعادته للمستخدم User1. ثم لنفترض أن المستخدم User2 يقوم بإرسال طلب GET للحصول على محتوى وليكن Content1 ليستقبل الطلب الموجه(2)، ويقوم بتنفيذ السيناريو السابق ذاته ويكون المحتوى مخزن في الموجه(3)، حيث يرسل الموجه(3) المحتوى باتجاه الموجه(2)، ليخزن ضمنه أيضاً في حال تحققت شروط الخوارزمية كـ عدد النُسخ والانشغالية، ومن ثم يرسل الموجه(2) المحتوى باتجاه User2. أيضاً لنفترض أن المستخدم User3 يقوم بإرسال طلب GET للحصول على محتوى وليكن Content1 ليستقبل الطلب الموجه (8)، ويقوم بتنفيذ السيناريو السابق ذاته ويكون المحتوى مخزن في الموجه (3)، حيث يرسل الموجه (3) المحتوى باتجاه الموجه (6)، لكن لا يُخزن ضمنه لأن عدد النُسخ أصبح أكثر من نسخة وبمكان قريب من بعضهما، ثم يرسل الموجه (6) المحتوى باتجاه الموجه (8)، ومن ثم باتجاه User3.

في ماسبق تم شرح سيناريو طلب المعلومات ذاتها، أما في حال طلب معلومات غير مخزنة بشكل مؤقت، فأنه يتم تنفيذ الخوارزمية بشكل مشابه لطلب المستخدم User1، كما أن الاتصال بين الموجهات يعتبر اتصال سريع جداً، لأنه من نوع اتصال ضوئي ولا يوجد توصيلات وسيطة، حيث الاستعلام عن معلومة في موجه آخر لايستهلك وقتاً طويلاً، لأنه مجرد جلب معلومة دون نقل المحتوى مثل حساب عدد النُسخ وتحديد انشغالية الموجهات.

8. تقييم الأداء



الشكل(4) مخطط الـ Sequence Diagram لتنفيذ التجربة في Icarus

تم تقييم أداء خوارزمية الحل المقترح CRJ من خلال المحاكي Icarus الإصدار v0.8.0، هو محاكي لـ ICN

قائم على الأحداث تم تنفيذه بلغة Python؛ وتم تطويره خصيصاً لتقييم أداء سياسات إدارة الذاكرة المخبئية وبروتوكولات التوجيه في شبكات الـ ICN. يستخدم نموذج تصميم MVC (Model – View –Controller). حيث يقوم الـ Model بتنفيذ الوظائف الأساسية لـ ICN، أما الـ View يراقب التغييرات على نموذج الشبكة، والـ Controller يعالج الأحداث والاستجابات، وتكون النتائج على أساس نموذج وبنية الشبكة [23].

8.1–سيناريو الاختبار: تم تضمين الحل المقترح CRJ إلى المحاكي Icarus، وذلك من خلال إضافة استراتيجية تحوي على الرماز البرمجي بلغة Python، ضمن *"crj.py\\strategy\\models\\icarus"* من أجل أن يتم مقارنة الحل المقترح مع البروتوكولات والسياسات المضمنة مسبقاً ضمن المحاكي، ليتم اسقاط الحلول كافة من بروتوكولات وسياسات لـ ICN بما فيها الحل المقترح على طبولوجيا الشبكة المضمنة في المحاكي Icarus كما الجدول(3)، وتم اختيار سياسة الاستبدال FIFO (First In First Out) حيث تتعامل هذه السياسة في استبدال المحتوى على مبدأ الرتل، من خلال تحرير المحتوى المُخزن أولاً واستبداله بالأحدث؛ أما عدد العُقد فهو متغير حسب كل طبولوجيا من طبولوجيات الشبكات المضمنة في المحاكي، ويتراوح بين الـ 30 عُقدة والـ 260 عُقدة.

كما يمثل الشكل(4) مخطط تنفيذ التجربة حيث المحرك Engine هو المسؤول عن إنشاء ذاكرة التخبئة واستراتيجية التوجيه، أما الاستراتيجة Strategy تحوي منطق التنفيذ لتوجيه الطلبات والمحتويات وتخزن بشكل مؤقت في ملف شبكة الاتصال، بينما الشبكة Network تمثل شبكة المحاكاة التي ستجري عليها التجربة، وأخيراً يقوم مجمع البيانات DataCollector بتلقي الأشعارات حول كل حدث في الشبكة ليفصّل كل حدث ويرسله لجامعي البيانات ويصدر في النهاية قيم معايير القياس مثل التأخير ومعدل إصابة الذاكرة المخبئية وغيرها من المعايير. Icarus يوفر عدة طوبولوجيا للإختبار موجودة في Topology Zoo. وهي 5 طبولوجيا كما الجدول (3)، مع استراتيجية استبدال ذاكرة التخزين المخبئي وهي: LRU ،LFU، FIFO، [23] RANDOM.

الجدول (3) سياسات التخبئة وبروتوكولات التوجيه في Icarus

Topologies	Strategies
WIDE GEANT GARR TISCALI ROCKET_FUEL	**CRJ:** Proposed solution **LCE:** Leave Copy Everywhere **LCD:** Leave Copy Down **CL4M:** Cache less for more **PROB_CACHE:** ProbCache **HR_SYMM:** Symmetric has routing h- **HR_ASYMM:** Asymmetric hash-routing **HR_MULTICAST:** Multicast hash-routing **HR_HYBRID_AM:** Hybrid Asymm-Multicast hash-routing **HR_HYBRID_SM:** Hybrid Symm-Multicast hash-routing

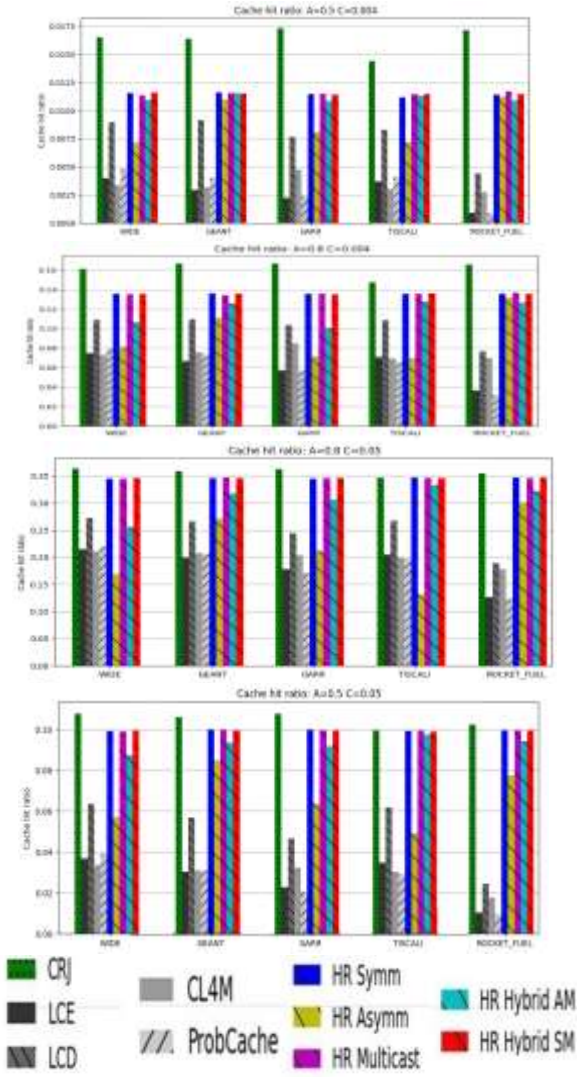
الجدول (4) معاملات المحاكاة

المُعامل	القيمة
Zipf distribution alpha value	[0.5 – 1]
عدد المحتويات Number of contents	300,000
حجم الذاكرة المخبئية Cahce Size	C=[0.002 – 0.05] * number of contents
عدد الطلبات في الثانية	12
عدد الطلبات التي تم إنشاؤها مسبقاً لملء الذاكرة المخبئية	300,000
عدد الطلبات التي يتم إنشاؤها لاحقاً من أجل القياس	600,000
Internal link delay (بين الموجهات فقط)	2 ms
External link delay (بين الموجهات والعُقد الأخرى)	34 ms
سياسة الاستبدال	FIFO

كما يتم توليد المحتوى من خلال تابع التوزيع Zipf حيث يحدد هذا طريقة توليد طلبات المحتوى وتكرارها في

التجربة مع المعامل alpha؛ وحجم الذاكرة المخبئة هو نسبة مئوية من العدد الكلي للمحتويات؛ ويوضح الجدول(4) مُعاملات المحاكاة لـ Icarus.

8.2-معايير تقييم الأداء: يفرض الحل المقترح CRJ الوصول لتصميم يحوي على سياسة تخبئة مع بروتوكول توجيه يحقق بنية ICN متكاملة، تساعد في تسريع الحصول على المعلومات من جهة المستخدم النهائي، وتخفيف الازدحام في طبولوجيا الشبكة؛ يتحقق ذلك من خلال زيادة فعالية الذاكرة المخبئية بشكل أساسي أي زيادة معدل الإصابة على الذاكرة Cache hit ratio، لأن ذلك سوف يقلل زمن الحصول على المعلومات عند طلبها من المستخدم النهائي كما يخفف الازدحام والحمل على الشبكة نتيجة تقليل عدد قفزات الحصول على المعلومات وجلبها من عُقدة تخبئة قريبة للمستخدم، حيث يتبع ذلك لطريقة إدارة الذاكرة المخبئية وتخزين المعلومات بما يناسب اهتمامات المستخدمين، وبالتالي لابد من قياس معيارين أساسيين في الشبكات المتمحورة حول المعلومات وهما كل من الـ Latency وذلك من أجل قياس التأخير في الشبكة، والـ Link load وذلك من أجل قياس الازدحام والحمل على الشبكة. وفي النهاية يتم محاكاة كافة البروتوكولات والسياسات ومن ثم قياس المعايير الثلاثة من أجل المقارنة فيما بينها حيث يكون الأفضل هو الأعلى Cache hit ratio والأقل Latency والأقل أيضاً Link load.



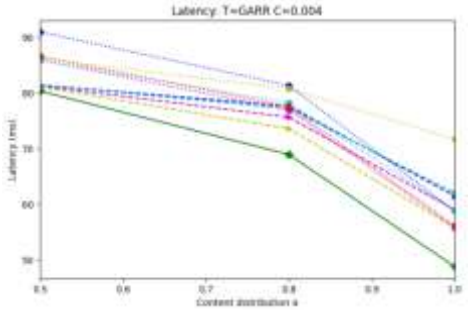
الشكل(5) مقارنة معدلات إصابة الذاكرة المخبئية C=0.004، C=0.05

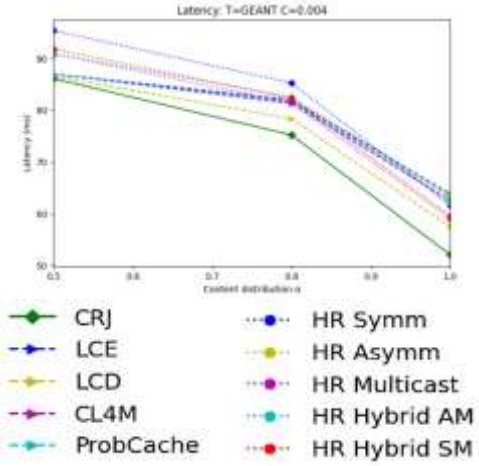
8.3-دراسة النتائج وتحليلها:

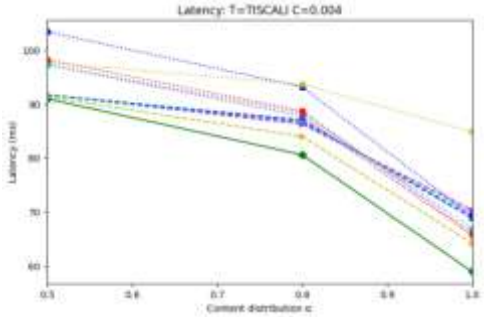
8.3.1- Cache hit ratio: تمت المحاكاة على 5 أنواع من طبولوجيا الشبكات المختلفة ولكل نوع 10 استراتيجيات في ICN، ونلاحظ اختلاف النتائج عندما alpha=0.5 وalpha=0.8 لأن ذلك يتبع شعبية المحتوى (طريقة توليد طلبات المحتوى وتكرارها). يظهر المخطط الإحصائي في الشكل(5) معدلات إصابة الذاكرة المخبئية عندما يكون حجم الذاكرة 0.004 من الحجم

الكلي للمحتويات وتكون alpha 0.5, 0.8 التي تتبع التوزيع Zipf، ويكون معدل إصابة الذاكرة المخبئية Cache hit ratio للحل المقترح CRJ أعلى من الاستراتيجيات الأخرى حيث وجدت في الذاكرة المخبئية حوالي 17% من عدد الطلبات الكلية في التجربة. كما يظهر الشكل(5) معدلات إصابة الذاكرة المخبئية عندما يكون حجم الذاكرة 0.05 و0.8 ,0.5 alpha حيث يكون معدل إصابة الذاكرة المخبئية Cache hit ratio للحل المقترح CRJ أعلى من الاستراتيجيات الأخرى حيث وجدت في الذاكرة المخبئية حوالي 37% من عدد الطلبات للتجربة.

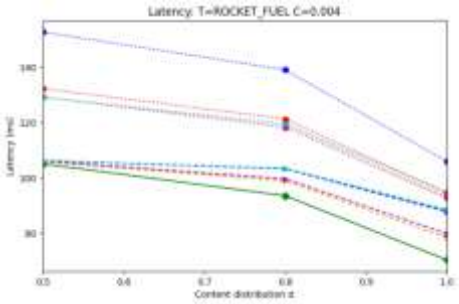
8.3.2- Latency: تمت المحاكاة على 5 أنواع من طبولوجيا الشبكات المختلفة ولكل نوع 10 استراتيجيات في ICN وتكون قيمة alpha بين الـ 0.5 والـ 1، ويتم حساب قيم التأخير لكل استراتيجية من خلال المعدل الوسطي لأزمنة التأخير لكل الطلبات ضمن كل جلسة في التجربة من أجل كل استراتيجية (مجموع كافة أزمنة التأخير لكل الجلسات / عدد الجلسات)؛ يظهر المخطط الإحصائي في الشكل(6) و (7) معدلات التأخير؛

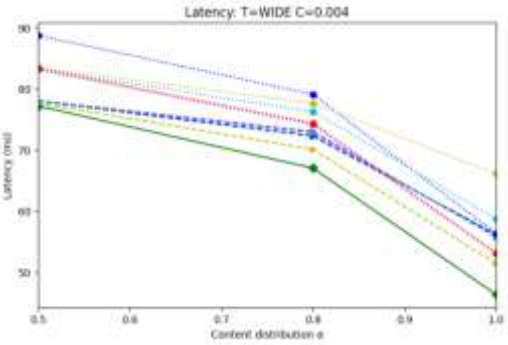




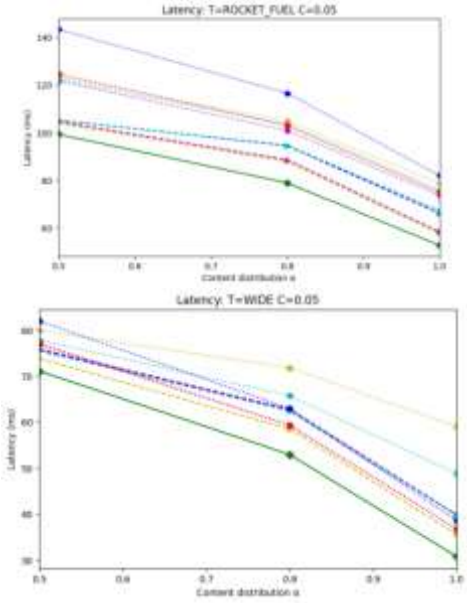


حيث حجم الذاكرة 0.004 و0.05 من الحجم الكلي للمحتويات ويكون معدل تأخير CRJ الأقل بين باقي الاسترتيجيات ويكون 30 ms عندما C=0.05، وكذلك الأقل في باقي التجارب وتنوع الطبولوجيا.





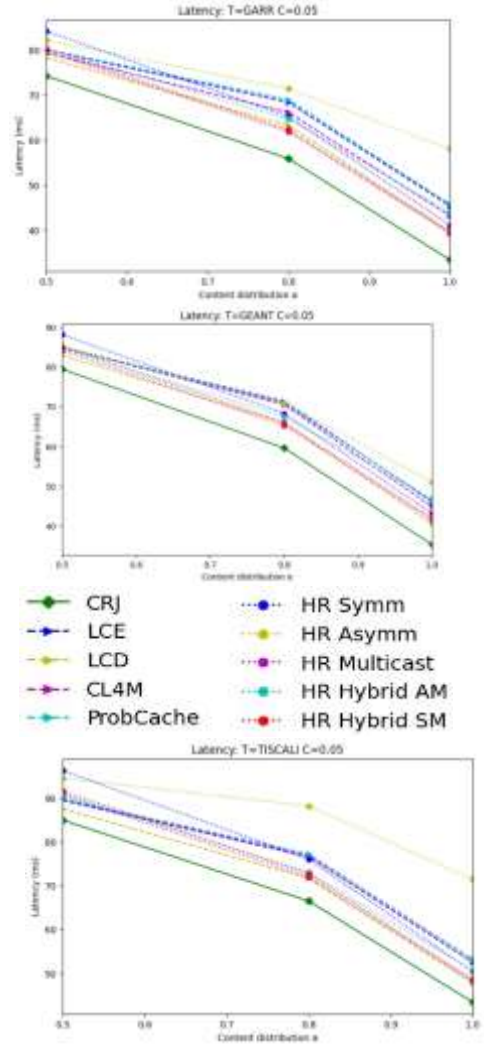
الشكل(6) مقارنة معدلات التأخير C=0.004

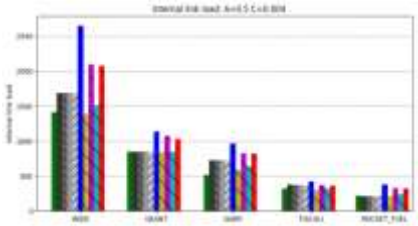


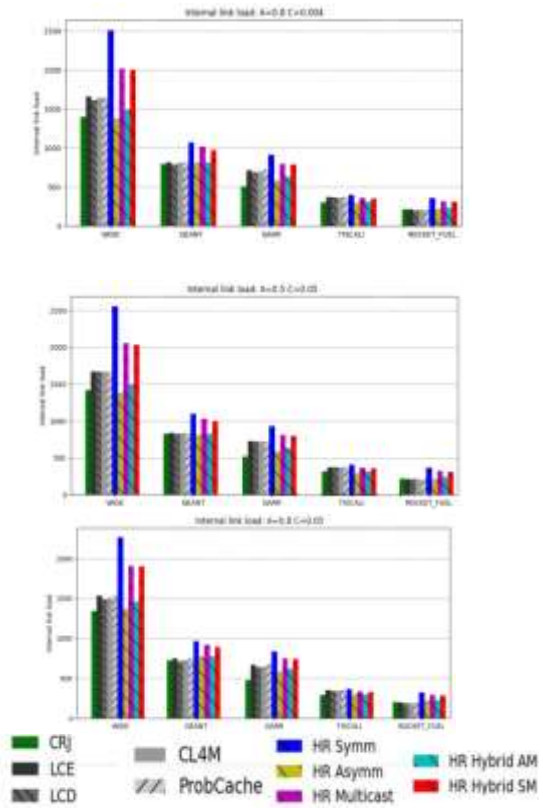
الشكل(7) مقارنة معدلات التأخير C=0.05

8.3.3- Link load: يظهر المخطط الإحصائي في الشكل(8) معدلات الحمل على المسار عندما يكون حجم الذاكرة 0.004 و 0.05 من الحجم الكلي للمحتويات، ويتم حساب قيم الحمل على المسار فقط بين موجهات التخبئة (Internal link load) وذلك من خلال المعدل الوسطي لحجم البيانات المتبادل لكل الطلبات ضمن كل جلسة في التجربة، من أجل كل استراتيجية هو:

(مجموع كافة احجام البيانات لكل الجلسات / الزمن الكلي للجلسات)؛ ويكون معدل الحمل على المسار link Internal load لـــ CRJ أقـــل أو قريـــب جـــداً مـــن الاستراتيجيات الأخرى. وبالتالي CRJ يقلل الازدحام في الشبكة ومع اختلاف الطبولوجيا يكون CRJ مناسب ويقلل الحمل على المسار.







الشكل(8) مقارنة معدلات الحمل على المسار C=0.004، C=0.05

9. الخلاصة وآفاق مستقبلية

تم تطوير Caching and Routing Jointly (CRJ) لدمج عمل سياسة إدارة الذاكرة المخبئية مع خوارزمية التوجيه، لتحقيق أهداف مشتركة ترفع كفاءة عمل ICN، حيث تم تطوير خوارزمية CRJ من خلال نموذج يتمحور حول المعلومات، ويقوم بنشر المعلومات ضمن الشبكة وأقرب مايمكن للمستخدم النهائي، وذلك لتقليل التأخير في معالجة الطلبات اللاحقة، وكذلك توزيع الأحمال في التخبئة على كافة عُقد الشبكة، وأيضاً تقليل عدد النُسخ من المحتوى المخبئ، وذلك من أجل ضمان تخبئة أكبر كمية من المعلومات المختلفة داخل عُقد الشبكة؛ كما تم التوجيه في CRJ من خلال تحديد المسار الأقصر وجلب المعلومات من العُقدة الأقرب، وفي حال كان تموضع المعلومات لايناسب المستخدم النهائي، عندها يتم تخزين المعلومات في عُقدة أقرب له، وذلك لتسريع تلبية الطلبات اللاحقة.

أما في الأعمال المستقبلية يمكن تحسين أساليب التخزين في الذاكرة المخبئية وإدارته بشكل ديناميكي، وذلك من خلال خوارزميات ترصد التغيرات في الشبكة. كما يمكن تطوير نموذج يساعد في تحديد عدد النُسخ المتماثلة ضمن الشبكة، وذلك اعتماداً على حاجات المستخدمين لمعلومات ذات أهمية. ومن الممكن أيضاً تحقيق نهج التخبئة الاستباقية في عُقد محددة من الشبكة، وذلك من خلال التنبؤ مستقبلاً بطبيعة طلبات المستخدم، ويمكن تحقيق ذلك من خلال خوارزميات تعمل على تحليل الطلبات الحالية، وتقوم بنقل المحتوى باتجاه عُقد أخرى بعيدة.

هذا البحث ممول من جامعة دمشق وفق رقم التمويل (501100020595).

References

[1] George Xylomenos *et al*., "A Survey of Information Centric Networking Research ", IEEE, 2013.

[2] Anand Seetharam *et al*., "On Caching and Routing in Information Centric Networks ", IEEE, 2017.

[3] Weibo Chu *et al.,* "Jointly Optimizing Throughput and Content Delivery Cost Over Lossy Cache Network", IEEE, 2021.

[4] Ngoc-Thanh Dinh, Younghan Kim, "An Efficient Distributed Content Store-Based Caching Policy for Information-Centric Networking" , MDPI, Basel, 2021.

[5] Krishna Delvadia, Nitul Dutta, and Rajendrasinh Jadeja, "CCJRF-ICN: A Novel Mechanism for Coadjuvant Caching Joint Request Forwarding in Information Centric Networks", IEEE, 2021.

[6] Quang N. Nguyen *et al*., "Adaptive Caching for Beneficial Content Distribution in Information-Centric Networking", IEEE, 2020.

[7] G. Zhang, Y. Li, and T. Lin, "Caching in Information Centric Networking: A Survey," Computer Networks, vol. 57, no. 16Nov. 2013, pp. 3128–41.

[8] Wang, S.; Zhang, X.; Zhang, Y.; Wang, L.; Yang, J.; Wang, W. A survey on mobile edge networks: Convergence of computing, caching and communications. IEEE Access 2017, 5, 6757–6779.

[9] W. K. Chai *et al*., "Cache Less for More in Information-Centric Networks," Proc. IFIP Networking, 2012.

[10] Fang, C.; Yao, H.; Wang, Z.; Wu, W.; Jin, X.; Yu, F.R. A survey of mobile information-centric networking: Research issues and challenges. IEEE Commun. Surv. Tutor. 2018, 20, 2353–2371.

[11] C. Giovanna, L. Mekinda, and L. Muscariello. "LAC: Introducing Latency-Aware Caching in Information-Centric Networks," Proc. LCN, 2015.

[12] K. Suksomboon *et al*. , "Popcache: Cache More or Less Based on Content Popularity for Information-Centric Networking," Proc. IEEE LCN, 2014.

[13] M. Badov *et al*., "Congestion- Aware Caching and Search in Information-Centric Networks," Proc. ACM ICN, 2014

[14] S. Eum *et al*., "CATT: Potential Based Routing with Content Caching for ICN," Proc. ACM ICN, 2012.

[15] L. Saino, I. Psaras, and G. Pavlou, "Hash-Routing Schemes for Information Centric Networking," Proc. ACM ICN, 2013.

[16] E. Rosensweig and J. Kurose "Breadcrumbs: Efficient, Best-Effrt Content Location in Cache Networks," Proc. IEEE INFOCOM, 2009.

[17] Sen Wang Jun and Bi Jianping Wu, "Collaborative Caching Based on Hash-Routing for Information-Centric Networking", Beijing, China, 2015.

[18] B. Banerjee *et al*., "Characteristic Time Routing in Information Centric Networks" Computer Networks, vol. 113, Feb.2017, pp. 148–58.

[19] S. Tarnoi, W. Kumwilaisak, and J. Yusheng, "Optimal Cooperative Routing Protocol Based on Prefi Popularity for Content Centric Networking," Proc. IEEE LCN, 2014.

[20] M.Zhang, H Luo and H Zhang, A Survey of Caching Mechanisms in Information-Centric Networking, IEEE Communications Surveys & Tutorials, 2015, 17(3), 1473–1499.

[21] Baixiang Huang *et al*., "Caching Joint Shortcut Routing to Improve Quality of Service for Information-Centric Networking", MDPI, 2018.

[22] J. Kurose,Information−centric networking :the evolution from circuits to packets to content, Elsevier Comput. Netw. 66 (2014) 112–120.

[23] L. Saino, I. Psaras, G. Pavlou, Icarus: a caching simulator for information centric networking (ICN), SimuTools (2014) 66–75.