# مجلة جامعة حمص

## سلسلة العلوم الهندسية الميكانيكية والكهربائية والمعلوماتية


**مجلة علمية محكمة دورية**

المجلد 47 . العدد 13

1447 هـ - 2025 م

| |
|---|
| **الأستاذ الدكتور طارق حسام الدين رئيس جامعة حمص**<br><br>**المدير المسؤول عن المجلة** |

| | |
|---|---|
| **أ. د. وليد حمادة** | **رئيس تحرير مجلة جامعة حمص للعلوم الإنسانية** |
| **د.نعيمة عجيب** | **رئيس تحرير مجلة جامعة حمص للعلوم الطبية والهندسية والأساسية والتطبيقية** |

| | |
|---|---|
| د.محمد فراس رمضان | عضو هيئة التحرير |
| د. مضر سعود | عضو هيئة التحرير |
| د. ممدوح عبارة | عضو هيئة التحرير |
| د. موفق تلاوي | عضو هيئة التحرير |
| د. طلال رزوق | عضو هيئة التحرير |
| د. أحمد الجاعور | عضو هيئة التحرير |
| د. الياس خلف | عضو هيئة التحرير |
| د. روعة الفقس | عضو هيئة التحرير |
| د. محمد الجاسم | عضو هيئة التحرير |
| د. خليل الحسن | عضو هيئة التحرير |
| د. هيثم حسن | عضو هيئة التحرير |
| د. أحمد حاج موسى | عضو هيئة التحرير |

تهدف المجلة إلى نشر البحوث العلمية الأصيلة، ويمكن للراغبين في طلبها الاتصال بالعنوان التالي:

**رئيس تحرير مجلة جامعة حمص**

**سورية . حمص . جامعة حمص . الإدارة المركزية . ص . ب (77)**

**. هاتف / فاكس : 2138071 31 963 ++**

**. موقع الإنترنت : www.homs-univ.edu.sy**

**. البريد الالكتروني : journal.homs-univ.edu.sy**

**ISSN: 1022-467X**

# شروط النشر في مجلة جامعة حمص

**الأوراق المطلوبة:**

- 2 نسخة ورقية من البحث بدون اسم الباحث / الكلية / الجامعة) + CD / word من البحث منسق حسب شروط المجلة.
- طابع بحث علمي + طابع نقابة معلمين.
- **اذا كان الباحث طالب دراسات عليا:**

يجب إرفاق قرار تسجيل الدكتوراه / ماجستير + كتاب من الدكتور المشرف بموافقته على النشر في المجلة.

- **اذا كان الباحث عضو هيئة تدريسية:**

يجب إرفاق قرار المجلس المختص بإنجاز البحث أو قرار قسم بالموافقة على اعتماده حسب الحال.

- **اذا كان الباحث عضو هيئة تدريسية من خارج جامعة البعث :**

يجب إحضار كتاب من عمادة كليته تثبت أنه عضو بالهيئة التدريسية و على رأس عمله حتى تاريخه.

- **اذا كان الباحث عضواً في الهيئة الفنية :**

يجب إرفاق كتاب يحدد فيه مكان و زمان إجراء البحث ، وما يثبت صفته وأنه على رأس عمله.

- **يتم ترتيب البحث على النحو الآتي بالنسبة لكليات (العلوم الطبية والهندسية والأساسية والتطبيقية):**

عنوان البحث . . ملخص عربي و إنكليزي ( كلمات مفتاحية في نهاية الملخصين).

1- مقدمة

2- هدف البحث

3- مواد وطرق البحث

4- النتائج ومناقشتها .

5- الاستنتاجات والتوصيات .

6- المراجع.

- **يتم ترتيب البحث على النحو الآتي بالنسـبة لكليات ( الآداب – الاقتصـاد– التربية – الحقوق – السياحة – التربية الموسيقية وجميع العلوم الإنسانية):**
- عنوان البحث . . ملخص عربي و إنكليزي ( كلمات مفتاحية في نهاية الملخصين).

1. مقدمة.
2. مشكلة البحث وأهميته والجديد فيه.
3. أهداف البحث و أسئلته.
4. فرضيات البحث و حدوده.
5. مصطلحات البحث و تعريفاته الإجرائية.
6. الإطار النظري و الدراسات السابقة.
7. منهج البحث و إجراءاته.
8. عرض البحث و المناقشة والتحليل
9. نتائج البحث.
10. مقترحات البحث إن وجدت.
11. قائمة المصادر والمراجع.

7- يجب اعتماد الإعدادات الآتية أثناء طباعة البحث على الكمبيوتر:

أ- قياس الورق 17.5×25 B5.

ب- هوامش الصفحة: أعلى 2.54- أسفل 2.54 – يمين 2.5- يسار 2.5 سم

ت- رأس الصفحة 1.6 / تذييل الصفحة 1.8

ث- نوع الخط وقياسه: العنوان . Monotype Koufi قياس 20

ـ كتابة النص Simplified Arabic قياس 13 عادي ـ العناوين الفرعية Simplified Arabic قياس 13 عريض.

ج . يجب مراعاة أن يكون قياس الصور والجداول المدرجة في البحث لا يتعدى 12سم.

8- في حال عدم إجراء البحث وفقاً لما ورد أعلاه من إشارات فإن البحث سيهمل ولا يرد البحث إلى صاحبه.

9- تقديم أي بحث للنشـر في المجلة يدل ضـمناً على عدم نشـره في أي مكان آخر، وفي حال قبول البحث للنشر في مجلة جامعة البعث يجب عدم نشره في أي مجلة أخرى.

10- الناشر غير مسؤول عن محتوى ما ينشر من مادة الموضوعات التي تنشر في المجلة

11- تكتب المراجع ضمن النص على الشكل التالي: [1] ثم رقم الصفحة ويفضل استخدام التهميش الإلكتروني المعمول به في نظام وورد WORD حيث يشير الرقم إلى رقم المرجع الوارد في قائمة المراجع.

**تكتب جميع المراجع باللغة الانكليزية (الأحرف الرومانية) وفق التالي:**

**آ . إذا كان المرجع أجنبياً:**

الكنية بالأحرف الكبيرة ـ الحرف الأول من الاسم تتبعه فاصلة ـ سنة النشر ـ وتتبعها معترضة ( – ) عنوان الكتاب ويوضع تحته خط وتتبعه نقطة ـ دار النشر وتتبعها فاصلة ـ الطبعة ( ثانية . ثالثة ) . بلد النشر وتتبعها فاصلة . عدد صفحات الكتاب وتتبعها نقطة.

وفيما يلي مثال على ذلك:

–MAVRODEANUS, R1986– **Flame Spectroscopy**. Willy, New York, 373p.

**ب . إذا كان المرجع بحثاً منشوراً في مجلة باللغة الأجنبية:**

ـــ بعد الكنية والاسم وسنة النشر يضاف عنوان البحث وتتبعه فاصلة، اسم المجلد ويوضع تحته خط وتتبعه فاصلة ـــ المجلد والعدد ( كتابة مختزلة ) وبعدها فاصلة ـــ أرقام الصفحات الخاصة بالبحث ضمن المجلة.

مثال على ذلك:

BUSSE,E 1980 Organic Brain Diseases **Clinical Psychiatry News** , Vol. 4. 20 – 60

**ج. إذا كان المرجع أو البحث منشوراً باللغة العربية فيجب تحويله إلى اللغة الإنكليزية و التقيد بالبنود (أ و ب) ويكتب في نهاية المراجع العربية: ( المراجع In Arabic )**

# رسوم النشر في مجلة جامعة حمص

1. **دفع رسم نشر (50000) ل.س أربعون ألف ليرة سورية عن كل بحث لكل باحث يريد نشره في مجلة جامعة البعث.**
2. **دفع رسم نشر (200000) ل.س مئة الف ليرة سورية عن كل بحث للباحثين من الجامعة الخاصة والافتراضية .**
3. **دفع رسم نشر (200) مئتا دولار أمريكي فقط للباحثين من خارج القطر العربي السوري .**
4. **دفع مبلغ (15000) ل.س ستة آلاف ليرة سورية رسم موافقة على النشر من كافة الباحثين.**

# المحتوى

# تقليل استهلاك الموارد في الشبكات ضمن الرقاقة غير المتزامنة القائمة على منطق الاتفاقية الفارغة NCL

اعداد الطالبة : رهف المصطفى اشراف : أ.د عبد الله غندور ، د. نبيل دحدوح

**الملخص**

مع زيادة حجم وتعقيد الأنظمة ضمن الرقاقة (SoC)، تُعد الشبكة ضمن الرقاقة (NoC) غير المتزامنة مرشحًا مناسبًا لإنشاء شبكة ربط بين مكونات النظام المتنوعة لما تتمتع به من مزايا تتراوح من زيادة كفاءة استهلاك الطاقة إلى تحسين استخدام المساحة وقابلية التوسع بالإضافة إلى التخلص من المشاكل المتعلقة بالمزامنة وتوزيع نبضة الساعة على كامل الرقاقة. تتألف هذه الشبكات من موجهات وخطوط نقل وواجهات ربط مع الكتل الوظيفية للنظام وتعتمد في تصميمها على تقنية الـ pipeline التي تستخدم على نطاق واسع ضمن البنى غير المتزامنة لتعزيز أداء الإنتاجية والتحكم بتدفق البيانات ضمن الشبكة في ظل غياب التزامن مع نبضة الساعة.

يعتبر منطق الاتفاقية الفارغة NCL ( Null Convention Logic ) أحد نماذج التصميم الواعدة في مجال تنفيذ بنى منطقية غير متزامنة أكثر استقراراً وأقل استهلاكاً للطاقة، لذلك كان هناك اهتمام متزايد مؤخراً لاستخدامه في تنفيذ الشبكات ضمن الرقاقة. ولكن مشكلته الأساسية تكمن في زيادة استهلاك الموارد مقارنة بنماذج التصميم الأخرى كونه يستخدم بوابات خاصة وترميز ثنائي الأسلاك.

نقدم ضمن هذا البحث آلية لتقليل استهلاك الموارد في الشبكات ضمن الرقاقة غير المتزامنة القائمة على المنطق NCL من خلال اقتراح بنية pipeline جديدة لتنفيذ موجه الشبكة بدلاً من NCL pipeline التقليدية. تجمع هذه البنية بين مزايا تقنية RL–NCL pipeline ( Register Less–NCL) والتصميم باستخدام بوابات العتبة NCL القياسية بحيث نستغني عن الحاجة لبوابات MTNCL ( Multi–threshold NCL ) المستخدمة في RL–NCL pipeline وبالتالي تصبح

قابلة للتحقيق على الـ FPGAs (Field Programmable Gate Arrays) أو باستخدام بوابات المنطق NCL القياسية.
تمت المحاكاة باستخدام أداة المحاكاة ®Cadence إلى جانب أداة التصميم بمساعدة الحاسب الأكاديمية VTR (Verilog-to-Routing) حيث بينت نتائج المقارنة انخفاضاً ملحوظاً في استهلاك الموارد عند تنفيذ موجه الشبكات ضمن الرقاقة باستخدام البنية المقترحة بنسبة 27.32 % مقارنة بالبنية المُنفذة باستخدام NCL pipeline التقليدية وهذا الانخفاض أدى بدوره إلى تحسن في زمن تأخير الموجه وتوفير استهلاك الطاقة.

الكلمات المفتاحية: دارات غير متزامنة، الشبكة ضمن الرقاقة، منطق الاتفاقية الفارغة، pipeline، FPGA، RL-NCL.

# Reducing Resource Consumption in Asynchronous Network-on-Chip Based on Null Convention Logic (NCL)

Abstract

As system on a chip (SoC) complexity grows, asynchronous networks on chip (NoC) emerge as effective solutions for interconnecting components, offering benefits like power efficiency, scalability, and elimination of synchronization issues. These networks consist of routers, links, and interfaces with system's functional blocks, and using pipeline technique, which is widely used in asynchronous architectures to enhance throughput performance and control data flow within the network in the absence of clock synchronization.

Recently, there has been increasing interest in using Null Convention Logic (NCL) for implementing Networks-on-Chip, as it is considered one of the promising design paradigms for constructing low-power robust asynchronous circuits. However, it suffers from increased resource consumption compared to other design paradigms, due to its use of specialized gates and dual-rail encoding.

This research introduces a resource-efficient method for implementing NCL-based NOC router architectures using Register-Less-NCL pipeline technique, and propose new method to design NCL gates, which eliminating the need for Multi-threshold NCL (MTNCL) gates in this pipeline, and can be implemented even in Field Programmable Gate

Arrays (FPGAs) or using the standard NCL cells method. Simulation results show that The new design of NOC router using the proposed RL-NCL pipeline architecture was able to achieve a significant reduction in resource consumption by 27,32% compared with the conventional NCL pipeline architecture. This reduction in resource consumption, in turn, leads to an improvement in router latency and power savings.

**Keywords**: Asynchronous Circuits, NOC, NCL, FPGA, Pipeline, RL-NCL.

## 1. مقدمة:

يزداد تعقيد الأنظمة الرقمية مع استمرار زيادة عدد الترانزستورات الموضوعة على رقاقة واحدة وفق قانون مور، حيث أصبح في الوقت الحاضر عدد الترانزستورات ضمن الرقاقة الواحدة في نطاق المليارات، كما يتم دمج المزيد من المكونات ضمن الرقاقة، مثل نوى المعالجة ومصفوفات الذواكر لتكوين الأنظمة ضمن الرقاقة SOC ( System on Chip )، وبالتالي أصبح التواصل بين هذه المكونات عاملاً محدداً لتحقيق أداء عالٍ واستهلاك منخفض للطاقة.

في بداية القرن العشرين ظهر مفهوم الشبكات ضمن الرقاقة NOC ( Network on Chip ) لحل مشكلة الاتصالات والتعقيد المتزايد ضمن الرقاقة، وهي مشابهة للشبكات المستخدمة في المجالات الأخرى حيث تتكون من عقد توجيه وقنوات تصل بينها، ولكن مع مرور الوقت أصبحت خاصية التزامن في الشبكات ضمن الرقاقة عاملاً مقيداً في بعض الأنظمة حيث تزداد صعوبة توزيع نبضات الساعة كلما أصبحت الأنظمة ضمن الرقاقة SOC أسرع ومعقدة أكثر ومساحتها أكبر [1] وهذا التحدي تواجهه أيضاً لوحات مصفوفات البوابات المنطقية القابلة للبرمجة FPGA (field programmable gate arrays) مما يجعل التصاميم غير المتزامنة (أو على الأقل التواصل بين الكتل الوظيفية ضمن الرقاقة بشكل غير متزامن) تأخذ حيزاً كبيراً من الاهتمام لأنها تخفف العبء الكبير الناتج عن توزيع نبضات الساعة الرئيسية. كما أنها توفر مزايا أخرى مثل انخفاض التداخل الكهرومغناطيسي EMI (electromagnetic interference) وانخفاض استهلاك الطاقة وإمكانية إعادة استخدام التصميم بالاعتماد على التشغيل الذاتي [2] .

هناك العديد من النماذج المستخدمة لتصميم البنى غير المتزامنة وهي تنقسم بشكل عام إلى مجموعتين نماذج التأخير المحدود BD (Bounded Delay) ونماذج غير حساسة للتأخير DI (Delay-Insensitive). في نماذج التأخير المحدود يتم تقييم التأخيرات في كل من البوابات والأسلاك وتحديدها وفقاً لأسوأ السيناريوهات لتجنب الأخطاء. من ناحية أخرى، لا تضع النماذج غير الحساسة للتأخير أي افتراضات حول التأخير في منطقها أو الترابط فيما بينها وتعتمد على بنيتها الداخلية وآلية ترميز البيانات للتحكم بتدفق البيانات دون حدوث أخطاء. في البداية كان الاهتمام موجه نحو نماذج التأخير المحدود كونها تستخدم بنية مشابهة إلى حد ما لبنية التصاميم المتزامنة ولكن مع تزايد مشاكل التوقيت والمسارات الحرجة أصبحت التصاميم غير الحساسة

للتأخير ملاذاً آمناً لبناء أنظمة ضمن الرقاقة أكثر استقراراً وتوفيراً للطاقة وخاصة في مجال الشبكات ضمن الرقاقة على الرغم من استهلاكها الكبير للموارد والمساحة مقارنة بالتصاميم الأخرى.

يعد منطق الاتفاقية الفارغة NCL [3] من أكثر النماذج الواعدة في مجال التصاميم غير الحساسة للتأخير كونه متكامل منطقياً (فهو ليس بحاجة لعناصر غير منطقية مثل الساعة أو عناصر تحكم أو خطوط تأخير لضبط وإكمال العملية)، و مستقل محليًا، متزامن ذاتيًا، وقادر على اكتشاف الأخطاء تلقائياً بالاعتماد على بنيته التركيبية.

نهدف ضمن هذا البحث إلى تحسين بنى الشبكات ضمن الرقاقة القائمة على المنطق NCL وتقليل المشاكل المتعلقة بهذا المنطق من خلال تعديل تقنية الـ pipeline المستخدمة ضمن هذه الشبكات لتصبح أقل استهلاكاً للموارد، وبناءاً على ذلك نستعرض بداية بنية الشبكات ضمن الرقاقة غير المتزامنة ومفهوم تقنية الـ pipeline ودورها ضمن هذه الشبكات و نستذكر خصائص المنطق NCL، وفي القسم الثاني نستعرض أبرز الدراسات المرجعية في مجال الـ pipeline، وأخيراً نقترح آلية لتحسين بنية الـ pipeline القائمة على المنطق NCL بهدف استخدامها في تحقيق شبكات ضمن الرقاقة أقل استهلاكاً للموارد وتم عرض نتائج تنفيذ موجه الشبكة باستخدام التقنية المقترحة ومقارنتها ببنية الموجه المُنفذ باستخدام NCL pipeline التقليدي في القسم الأخير من هذه الدارسة.

## 2. هدف البحث:

أغلب الأنظمة ضمن الرقاقة حالياً تستخدم الشبكات كوسيلة للتواصل بين مكونات النظام نظراً لأداءها الفعال من حيث الإنتاجية والموثوقية والأمان مقارنة بالوصلات التقليدية ولكنها تشكل عبء على النظام من ناحية استهلاك الموارد، يهدف هذا البحث إلى تقليل استهلاك الموارد من قبل الشبكات ضمن الرقاقة غير المتزامنة القائمة على المنطق NCL مع المحافظة على أداءها من خلال تحقيق بنية الموجه باستخدام Register–Less–NCL pipeline ولكن تصميم هذا الـ pipeline مخصص بالكامل حيث تم تنفيذه على مستوى الترانزستور بالاستفادة من بوابات المنطق MTNCL وبالتالي غير قابل للتنفيذ باستخدام بنى المنطق NCL القابلة لإعادة البرمجة

لذلك تم اقتراح بنية جديدة تجمع بين مزايا تقنية RL−NCL pipeline والتصميم باستخدام بوابات NCL القياسية.

### 3. الشبكات ضمن الرقاقة غير المتزامنة:

تتكون الشبكة ضمن الرقاقة من عقد التوجيه وقنوات تصل فيما بينها. تحدد هيكلية الشبكة ترتيب هذه العقد وكيفية ربطها ببعضها بواسطة القنوات، ويكون اختيار هيكلية الشبكة عادةً هو الخطوة الأولى في تصميم الشبكة حيث تعتمد استراتيجيات التوجيه وآلية التحكم بالتدفق بشكل كبير عليها، وهناك هيكليات عديدة للشبكات ضمن الرقاقة تختلف باختلاف الغرض من الرقاقة بهدف الحصول على أفضل أداء للشبكة.

تنقسم الشبكات ضمن الرقاقة حسب تصميمها إلى شبكات متزامنة بالكامل وشبكات متعددة التزامن وشبكات غير متزامنة. تتضمن الشبكة المتزامنة بالكامل ساعة واحدة موزعة على كامل شبكة الاتصالات والتي قد تعمل بمعدل ساعة مختلف عن الوحدات الوظيفية التي تربط بينها ونظراً لأن الشبكة تمتد عبر الرقاقة فأن توزيع نبضات الساعة يمثل تحدياً كبيراُ، فوفقاً للأبحاث تستهلك هذه الشبكات ما يصل إلى 30% من ميزانية طاقة الرقاقة بسبب توزيع الساعة [4]، كما أنها ذات معدل إنتاجية محدود وعادةً ما تكون أبطأ بـ 2−3 مرات من الوحدات الوظيفية مما يؤثر بشكل سلبي على أداء عملية نقل البيانات، أما في الشبكات متعددة التزامن يكون لكل موجه تردد تشغيل خاص يمكن أن يساوي تردد تشغيل الطرفية المتصل بها، مما يقلل من استهلاك الطاقة الديناميكي عند مقارنته بالشبكات أحادية التردد، ولكن يجب إضافة واجهات تزامن بين الموجهات وبين كل موجه والوحدة الوظيفية التي تعمل ضمن مجال تزامن مختلف.

بالنسبة للشبكات ضمن الرقاقة غير المتزامنة تستخدم أشارات المصافحة لضبط تدفق البيانات ضمن الشبكة بدلاً من تزامن نبضة الساعة مما يسمح بتكامل أكثر مرونة لمكونات النظام ذات خصائص المزامنة المختلفة، حيث يمكن أن تستخدم كشبكة اتصال لوحدات وظيفية غير متزامنة أو متزامنة كما في أنظمة GALS ((Globally Asynchronous Locally Synchronous مما يلغي الحاجة إلى إدارة الساعة عبر شبكة واسعة، ويقتصر التزامن على عقدة المصدر وعقدة الوجهة فقط وبالتالي تقلل التأخيرات الزمنية الناتجة عن واجهات المزامنة، إلى جانب تقديم استهلاك

منخفض للطاقة عند مقارنتها بالشبكات المتزامنة، حيث لا يتم استهلاك أي طاقة ديناميكية تقريباً عندما تكون الشبكة في وضع الخمول.

يختلف تصميم الشبكات غير المتزامنة باختلاف نموذج التصميم المستخدم الذي يحدد من خلاله آلية عمل برتوكول المصافحة وطريقة ترميز البيانات، ولكن جميع هذه التصاميم تعتمد آلية العمل نفسها تقريباً وهي أيضاً المتبعة في تصميم معظم الدارات الرقمية والتي تقوم على تقسيم الدارة إلى مراحل مع إدخال مسجلات بينها للاحتفاظ بالقيم الناتجة عن كل مرحلة وتمريرها للمرحلة التالية بهدف زيادة الإنتاجية وهذا ما يعرف باسم تقنية الـ pipeline .

### 4. تقنية الـ pipeline:

تستخدم الـ pipeline على نطاق واسع لتعزيز أداء الإنتاجية في الأنظمة الرقمية عالية السرعة مثل المعالجات الحديثة المستخدمة للأغراض العامة، ووحدات الرسوم، والدارات الرقمية الخاصة بتطبيقات معينة، مثل الوسائط المتعددة، وكذلك هو الحال بالنسبة للموجهات في معظم أنظمة الشبكات ضمن الرقاقة عالية الأداء. يقسم تنفيذ الـ pipeline النموذجي الوحدات الوظيفية المعقدة إلى وحدات أصغر، حيث تحقن رزم البيانات ضمن الـ pipeline على التوالي، ويتم معالجتها من قبل الوحدات الجزئية المختلفة بالوقت نفسه.

على الرغم من أن فكرة المعالجة المتوازية لرزم الدخل المتعددة هي نفسها، إلا أن الـ pipeline غير المتزامنة تتمتع بعدة مزايا أساسية مقارنة بالـ pipeline غير المتزامنة [5] أهمها:

(1) في الـ pipeline المتزامنة، يتم حساب تردد الساعة من خلال المسار الحرج للدارة وتعمل جميع المراحل بنفس المعدل، بينما في التصميم غير المتزامن، تعمل كل مرحلة وفقًا لمسارها الحرج الخاص بها، مما يقلل من زمن التأخير.

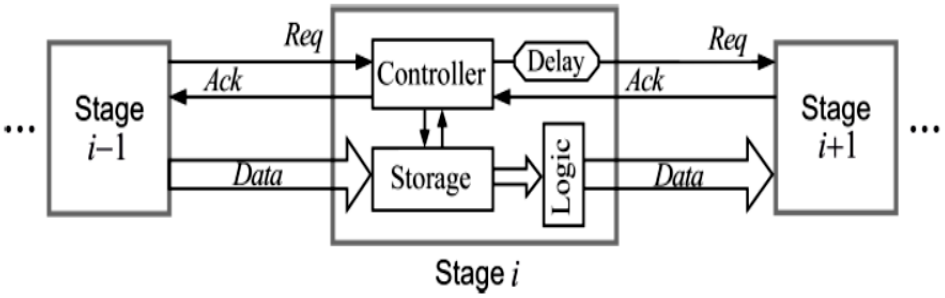
(2) يمكن أن تتلقى مراحل الـ pipeline غير المتزامنة عدد متغير من رزم البيانات في أي وقت، ولهذا السبب تتم معالجة كل من هذه الرزم عند توفرها، بينما في المتزامنة لضمان سير العملية بشكل صحيح يجب أن تصل البيانات الجديدة خلال فترات زمنية محددة مسبقًا.

(3) التحكم المتأصل في التدفق بسبب بروتوكولات المصافحة التي تستخدمها الأنظمة غير المتزامنة.

(4) يكون استهلاك الطاقة الديناميكية عند الطلب، أي فقط عندما يكون هناك بيانات يجب معالجتها وغير ذلك تكون الدارة خاملة.

وقد أدى هذا إلى العديد من المنتجات الناجحة مثل IBM–TrueNorth و IBM–FIR Filter و Achronix–Fastest FPGAs و Intel RAPPID ورقائق مبدلات Intel/Fulcrum Ethernet [6–9].

يبين الشكل (1) بنية توضيحية لتقنية الـ pipeline غير المتزامنة بشكلها الخطي، حيث ترتبط المراحل المتتالية بقناة اتصال تمتد عبر إشارات المصافحة، والتي تتكون عادةً من الطلب (Req) وتأكيد الطلب (Ack)، بالإضافة إلى البيانات (Data) التي يتم إرسالها.




**الشكل (1): بنية توضيحية لتقنية الـ pipeline غير المتزامنة.**

بشكل عام، تستقبل المرحلة i البيانات من المرحلة السابقة (المرحلة i–1)، ويتم تخزينها ومعالجتها قبل إرسالها إلى المرحلة اللاحقة (المرحلة i+1). وحدة التحكم داخل "صندوق المرحلة" هي التي تلعب دور تنسيق إشارات المصافحة بالإضافة إلى تنظيم التوقيتات المحلية لتخزين البيانات و/أو معالجتها.

افترضنا في البنية الموضحة في الشكل (1) استخدام نموذج نقل البيانات المجمعة ذات التأخير المحدود، حيث تم إضافة عنصر تأخير صريح إلى مسار الطلب لمطابقة أسوأ تأخير ضمن كتلة المنطق (Logic). إذا لم تكن هناك حاجة لجزء المنطق، فإن البنية تصبح عبارة عن نظام FIFO ( First–In–First–Out ) غير متزامن.

تجدر الإشارة إلى أن الدارة العملية تعتمد عادةً على كلٍّ من الهياكل الخطية وغير الخطية لتقنية الـ pipeline، مثل التفرعات والوصلات والمحكمات.

### 1.4. نظرة عامة على تنفيذ دارة الـ pipeline غير المتزامنة:

يتطلب التنفيذ الخاص لبنية الـ pipeline الموضح في الشكل (1) حل مجموعة من مشكلات التصميم، بدءًا من الاعتبارات عالية المستوى، مثل مخططات ترميز البيانات وبروتوكولات إشارات المصافحة، ووصولًا إلى خيارات مستوى الدارة، مثل عائلات المنطق وطرق تخزين البيانات، وذلك لتحقيق الأهداف المتعلقة بمساحة الدارة والسرعة واستهلاك الطاقة. في هذا القسم من البحث، نناقش هذه القضايا التصميمية من خلال دراسة قنوات الاتصال أولًا، ثم دراسة بنية مرحلة الـ pipeline.

#### 1.1.4. قنوات الاتصال:

##### أ. ترميز البيانات:

يعتمد تصميم الـ pipeline غير المتزامن بشكل رئيسي على الطريقة المستخدمة في ترميز البيانات، حيث هناك طريقتان رئيسيتان لترميز البيانات في الدارات غير المتزامنة هما (أ) الترميز ثنائي الأسلاك وهو المستخدم غالباً في نماذج التصميم غير الحساسة للتأخير [10–14] و(ب) ترميز البيانات المجمعة أحادية السلك المستخدم عادةً في نماذج التأخير المحدود [15–20]. في الترميز ثنائي الأسلاك، يتم استخدام سلكين لترميز كل بت (أحدهما للبيانات والآخر للإشارة إلى صحتها). من ناحية أخرى، يحافظ ترميز البيانات المجمعة أحادية السلك، كما هو موضح في الشكل (1)، على تطبيقات المنطق المتزامن أحادي السلك التقليدية المستخدمة ضمن مسار البيانات. ومع ذلك، فإنه يُدخل إشارات مصافحة إضافية للإشارة إلى صحة البيانات، كما يستخدم أيضاً تقنية تُعرف باسم مطابقة التأخير [21]، أي إدراج عنصر تأخير صريح ضمن مسار إشارة الطلب Req يتم تنفيذه عادةً بواسطة سلسلة عاكس أو نسخة طبق الأصل عن المسار الحرج ضمن كتلة المنطق لمطابقة أسوأ تأخير لهذه الكتلة ضمن المرحلة الواحدة.

بشكل عام، تتطلب الدارات ذات الترميز ثنائي الأسلاك مساحة إضافية. ومع ذلك، فهي تُمكّن من قياس التأخير في الوقت الفعلي من خلال استشعار حالة الأسلاك باستخدام دارات كشف الإكمال [21]، وبالتالي تتمتع بمتانة توقيت أكبر، بينما تُعدّ دارات البيانات المجمعة أسهل للمصممين ذوي التفكير المتزامن، على الرغم من أن هذه الدارات يجب أن تضمن هوامش كافية لتأخير مسار البيانات في أسوأ الحالات في ظل تغيرات PVT (process, voltage and temperature).

**ب. بروتوكولات المصافحة:**

يشير بروتوكول المصافحة إلى تسلسل انتقالات إشارات المصافحة لتسهيل نقل البيانات، حيث يشير الانتقال إما إلى الحافة الصاعدة (+) أو الحافة الهابطة (−) لإشارة Req أوAck. بروتوكولات إشارات المصافحة الرئيسية الثلاثة هي ثنائي الطور، ورباعي الطور، وأحادي المسار. بروتوكول المصافحة ثنائي الطور [22,20,15,13]، ويُعرف أيضًا باسم بروتوكول عدم العودة إلى الصفر (NRZ) لأن كل انتقال لـ Req وAck ينقل رسالة نشطة، على سبيل المثال، يُشير كلٌّ من +Req و−Req إلى أن البيانات المُجمّعة أصبحت صالحة، بينما يُشير كلٌّ من +Ack و-Ack إلى أن البيانات قد استُهلكت.

أما بالنسبة لبروتوكول المصافحة رباعي الطور [22,20,16]، أو بروتوكول العودة إلى الصفر (RZ) فهو يستخدم حافة واحدة فقط لكل إشارة مصافحة لنقل الرسائل النشطة، بينما يُعاد ضبط الحافة الأخرى ببساطة. تُمكّن التركيبات المختلفة للحواف النشطة لـ Req وAck من ظهور أنواع فرعية مختلفة من البروتوكولات رباعية الطور.

يدمج البروتوكول أحادي المسار[23] بين طلب الإرسال req والتأكيد ack ضمن سلك واحد حيث يُصدر المُرسِل طلبًا عن طريق إنشاء انتقال على السلك، ويُقرّ المُستقبِل بالاستلام عن طريق عكس هذا الانتقال. لذلك، يمتلك هذا البروتوكول ميزات كلٍّ من البرتوكولين السابقين (ثنائي ورباعي الطور).

فيما يتعلق بكفاءة الاتصال لبروتوكولات المصافحة المذكورة أعلاه، يُعتقد عمومًا أن البروتوكولات ثنائية الطور وأحادية المسار تُمكّن من معدل بيانات أعلى، أو زمن دورة أقصر لدارات الـ pipeline، مقارنةً بالبروتوكولات رباعية الطور [5]، ولكن قد تتطلب الدارات القائمة على بروتوكولات المصافحة ثنائية الطور دارات تحويل إضافية تتفاعل مع التخزين والمنطق القائم على المستوى، وقد تُسبب صعوبات في تصحيح الأخطاء [24].

### 2.1.4. مرحلة الـ pipeline ( تنفيذ تخزين البيانات والمنطق ووحدة التحكم ):

بعد تحديد مخطط ترميز البيانات وبروتوكول المصافحة، يحين الوقت المناسب للنظر في تنفيذ الدارة ضمن مرحلة الـ pipeline، حيث يتم بشكل عام اتباع تسلسل التصميم التالي المكون من ثلاث خطوات:

**1. تحديد عناصر تخزين البيانات:**

يعود اختيار عناصر التخزين إلى نموذج التصميم المستخدم حيث يمكن أن تكون عناصر تخزين البيانات المستخدمة ضمن الـ pipeline غير المتزامنة هي نفسها المعتمدة في الدارات التسلسلية المتزامنة، مثل المواسك الحساسة للمستوى والقلابات الحساسة للحافة، أو يمكن اختيار عناصر تخزين بيانات مصممة خصيصًا لبنى pipeline غير المتزامنة، مثل مواسك التقاط التمرير المستخدمة في الـ micro–pipeline.

**2. اختيار نمط المنطق:**

يمكن استخدام كلٍّ من نمطي المنطق الساكن والديناميكي لتنفيذ مسار البيانات ضمن بنى الـ pipeline غير المتزامنة التي تعتمد في تصميمها على نموذج التأخير المحدود. يتميز المنطق الساكن بأنه خالٍ من تداخل إشارات المصافحة، كما هو موضح في الشكل (1). أما المنطق الديناميكي، فيتطلب تنسيق مرحلتي الشحن المسبق والمعالجة بواسطة وحدات تحكم محلية، ولكنه قد يُحسّن أداء الدارة بشكل أكبر على حساب مخطط تحكم أكثر تعقيداً.

بالمقارنة مع نموذج التأخير المحدود، فإن تصميم مسار بيانات باستخدام النموذج غير الحساس للتأخير DI يتطلب جهدًا أكبر، إذ يجب تنفيذ شكل مناسب من المؤشرات للتحكم قي انتقال البيانات بين المراحل، إما عبر المنطق نفسه أو من خلال دارات كشف إكمال إضافية حيث توجد أشكال مختلفة للتصميم غير الحساس للتأخير مثل منطق الاتفاقية الفارغة (NCL) القائم على بوابات العتبة وتوليف الحد الأدنى غير الحساس للتأخير (DIMS). علاوة على ذلك، تتوفر تقنيات تحسين منطقي لتقليل تعقيد الدارة مع الحفاظ على الاستقرار.

**3. تصميم وحدة التحكم:**

تلعب وحدة التحكم بالمصافحة دورًا رئيسيًا في تنسيق إشارات المصافحة بين المراحل المتجاورة لـ pipeline وتعتبر من أكثر مجالات البحث اهتماماً في تصميم الدارات غير المتزامنة، حيث يجب أن يتبع سلوك وحدة التحكم المُنفَّذة بدقة علاقات التوقيت المنسّقة جيدًا مع البيئة،

وألا تُولِّد أي أعطال زمنية (ما يُسمى بالخالي من المخاطر) بخلاف كتلة المنطق ضمن مسار البيانات، والتي يمكن أن تتسامح بحدوث بعض الأعطال المؤقتة، كما يعتمد التشغيل الصحيح لدارة التحكم غير المتزامنة على افتراضات نموذج التأخير الخاص بها.

**5. منطق الاتفاقية الفارغة NCL :**

كما ذكرنا في مقدمة هذا البحث يمكن تصنيف الدارات غير المتزامنة وفقًا لنموذج تأخيرها إلى دارات ذات تأخير محدود (BD) ودارات غير حساسة للتأخير (DI). تعتمد الأولى في تصميمها على قيم تأخير محددة وفترات زمنية محدودة للعمل بشكل صحيح. إذا لم تلتزم الدارة بهذه الفترات، فلا يمكن ضمان التشغيل الصحيح. لذلك تُدرج ضمن الدارة تأخيرات مطابقة أكبر من تأخيرات المسارات الحرجة.

أما الدارات غير الحساسة للتأخير (DI) فهي تعمل بشكل صحيح بغض النظر عن تأخير البوابات والأسلاك. وبالتالي، يُعد نموذج DI الأكثر استقراراً من بين النماذج غير المتزامنة ولكن صعب التحقيق من الناحية العملية لذلك يعتبر النموذج شبه غير الحساس للتأخير (QDI) هو النموذج الأكثر استقرار الذي يُمكن تطبيقه في معظم الدارات. حيث تعمل دارات QDI بشكل صحيح بغض النظر أيضاً عن تأخير البوابات والأسلاك، ويقتصر القيد الزمني الوحيد على بعض تأخيرات الأسلاك، مثل افتراضات التفرعات المتزامنة والتغذية العكسية.

وبهدف تحقيق استقرار أكبر يتم استخدام رموز تدعم النموذج شبه غير الحساس للتأخير، مثل ترميز الأسلاك المزدوجة وهو الترميز الأكثر شيوعًا في تصميم دارات NCL، حيث يُمثل كل بت من البيانات بسلكين، سلك يعبر عن الصفر منطقي (أو False) وسلك يعبر عن الواحد منطقي (أو True) وفقاً للجدول (1) المبين أدناه حيث لا يمكن أن يكون كلا السلكين واحد منطقي في نفس الوقت، لذا يتم فصل الانتقال من قيمة بيانات صالحة إلى أخرى بقيمة NULL (قيمة فارغة).

**الجدول(1): التعبير وفق الترميز مزدوج الأسلاك.**

| $A^1$ | $A^0$ | A |
|---|---|---|
| 0 v | 0 v | NULL |
| 0 v | 1 v | Logic 0 (data) |
| 1 v | 0 v | Logic 1 (data) |
| 1 v | 1 v | Not allowed |

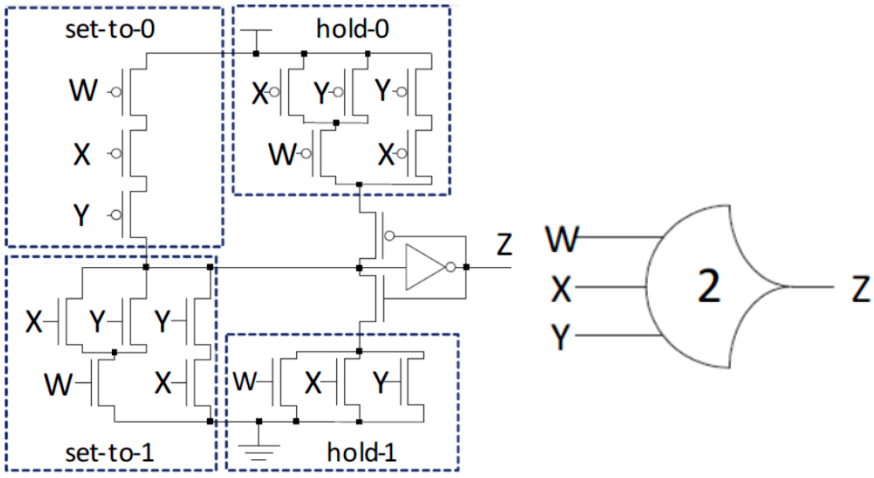
يمكن أن تستخدم الدارات غير المتزامنة كما ذكرنا سابقاً ضمن هذا البحث بروتوكول ثنائي الطور أو رباعي الطور للتعبير عن حالة معينة على أسلاكها، ونظرًا لطبيعة الدارات ثنائية الأسلاك فهي تستخدم عادةً بروتوكول رباعي الطور. إلا أن الوضع مختلف قليلاً بالنسبة لدارات NCL حيث يُشير الانتقال من منخفض إلى مرتفع إلى إقرار باستلام بيانات صالحة، ويُشير الانتقال من مرتفع إلى منخفض إلى إقرار بقيمة NULL.

منطق الاتفاقية الفارغة NCL [3] هو نموذج منطقي غير متزامن شبه حساس للتأخير (QDI) حيث يكون التحكم متأصلًا في طريقة التعبير عن البيانات، وبالتالي فهو يوفر تصميمات جاهزة للوصل والتشغيل، ولا يتطلب تحليلًا لأسوأ حالات التأخير. تم استخدام NCL بنجاح في عدد من المنتجات التجارية، بما في ذلك وحدات التحكم الدقيقة والمنتجات الطبية المضمنة ومحركات التشفير لتطبيقات البطاقات الذكية [26]. بالإضافة إلى ذلك، تم تطوير العديد من أدوات أتمتة التصميم الإلكتروني (EDA) الخاصة بالمنطق NCL [27]، وفي السنوات الأخيرة، تم استخدام NCL في مجموعة متنوعة من التطبيقات، بما في ذلك تصميم الدارات منخفضة الطاقة [28]، ودارات التشفير المقاومة للهجوم على الأخطاء [29].

تُنفذ دارات NCL باستخدام بوابات العتبة التي تعتبر العناصر الأولية لتحقيق منطق NCL، ويُرمز لها بالرمز THmn وتسمى بوابة العتبة m–of–n ، أي لها n مدخل وقيمة عتبة m حيث $n \geq m \geq 1$، وتتميز بقدرة الاحتفاظ بالحالة أي أنه بمجرد ضبط خرج البوابة THmn يتم الاحتفاظ بحالة الخرج هذه ولن يُعاد ضبطه إلا عند إعادة ضبط عدد m على الأقل من المداخل. وبالتالي

كما هو الحال في التصميم المتزامن الذي يستخدم بوابات المنطق البوليانى في تنفيذ الدارات، هنا يتم تحقيق دارات NCL باستخدام بوابات العتبة.

يوضح الشكل (2– أ) رمز بوابة العتبة TH23 كمثال عن بوابات العتبة ويبين الشكل (2– ب) بنية هذه البوابة على مستوى الترانزستور CMOS حيث تتألف من أربع كتل وظيفية هي: set–to–1 ، و set–to–0، و hold–1 و hold–0 بالإضافة إلى عاكس على الخرج مع تغذية عكسية.




–ب– –أ–

**الشكل (2): بوابة العتبة TH23: أ– رمزها – ب– بنيتها على مستوى الترانزستور.**

### 6. الدراسة المرجعية:

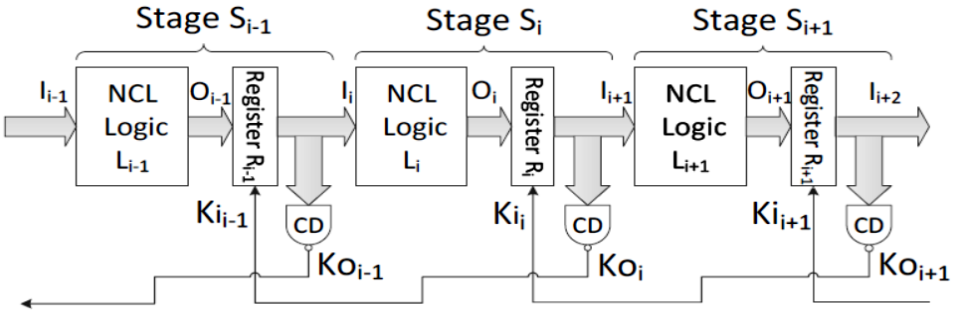
تم تنفيذ العديد من تقنيات الـ pipeline غير المتزامنة باستخدام نموذج التأخير المحدود وترميز البيانات المُجمعة أحادي الأسلاك مثل MOUSETRAP و GasP و look–ahead pipeline(LP) وتعديلاتها المستندة إلى Williams PS0 pipeline بالإضافة إلى pipeline EDP ( Embedded Delay Pipeline ) [15–20]. يستخدم MOUSETRAP مواسك شفافة قياسية ضمن مسار التحكم وبنى منطقية ثابتة لمسار البيانات الخاص به. أيضاً تم استخدام المواسك في GasP ولكنها تتطلب مسارات تحكم متوازنة وتحتاج إلى تلبية قيود التوقيت ثنائية

الجوانب. من ناحية أخرى، تتضمن $LP_{sr}$ pipeline مثل $LP_{sr}$ 2/2 و $LP_{sr}$ 2/1 مسار بيانات منطقي ديناميكي ولا يلزم وجود مواسك حيث يتم تحقيق إنتاجية أعلى في $LP_{sr}$ 2/1 على حساب زيادة تبديد الطاقة وتعقيد التصميم بسبب الحاجة إلى تلبية ثلاثة قيود توقيت. أما بالنسبة لـ pipeline ذو التأخير المُدمج EDP، فهو يتألف من عناصر تأخير مُدمجة، وذلك للتغلب على مشاكل الـ pipelines السابقة، حيث تعمل هذه العناصر كوحدة تحكم وعنصر تأخير مُطابق بنفس الوقت، وتتطلب استيفاء قيد توقيت واحد فقط.

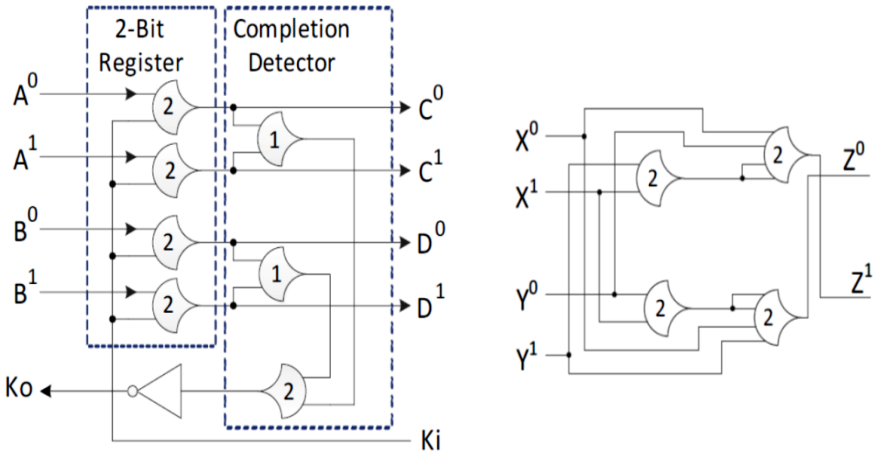
ركزت أبحاث أخرى على نماذج التصميم شبه غير حساسة للتأخير [10–14](حيث أن تحقيق البنى غير الحساسة للتأخير تماماً صعب من الناحية العملية)، حيث تم في [13] اقتراح بنية pipeline غير حساسة للتأخير تعتمد برتوكول ثنائي الطور مما يُتيح تقليل التأخير واستهلاك الطاقة مقارنةً ببنية QDI التي تستخدم بروتوكول رباعي الطور.

كما تم تنفيذ pipeline هجينة مثل الـ pipeline الهجين مع تأكيد الطلب المسبق (EA–Hybrid) والـ pipeline الهجين عالي السعة مع الكشف اللاحق (PD–Hybrid)[25] ، التي تستخدم مسارات بيانات هجينة تجمع بين متانة الترميز ثنائي الأسلاك وبساطة مخططات الترميز أحادية السلك.

يدعم التصميم القائم على المنطق NCL أيضاً تقنية الـ pipeline [30] ويبين الشكل (3) بنية NCL pipeline التقليدية ذات ثلاث مراحل. تتكون كل مرحلة من الجزء المنطقي NCL Logic (المُنفذ باستخدام بوابات العتبة)، ومسجلات NCL لتخزين القيم المرحلية، ودارة كاشف الإكمال CD (Completion Detector)، التي تشير إلى اكتمال عملية المعالجة وتوفر بيانات جديدة.




**الشكل (3): بنية NCL pipeline التقليدية.**




– ب – –أ–

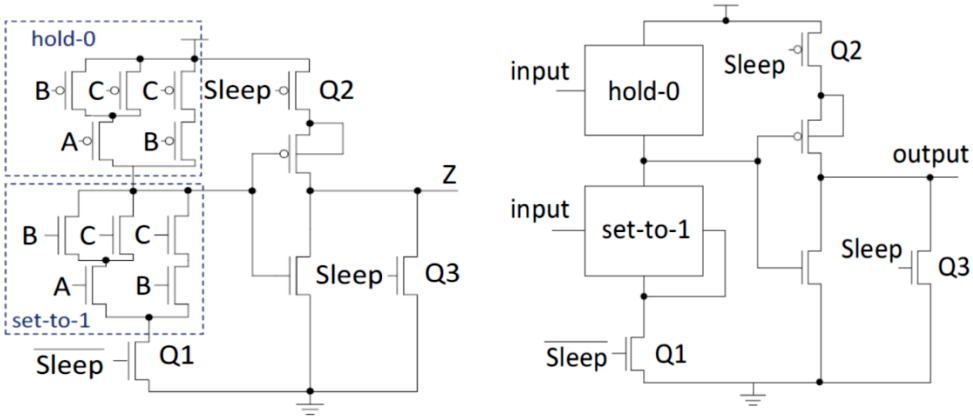
**الشكل (4): أ – تنفيذ دالة المنطق NCL المكافئة لبوابة XNOR بمدخلين – ب– بنية مسجل NCL وكاشف الإكمال CD الموافقين لها.**

تم استخدام بوابات العتبة لتنفيذ الجزء المنطقي، والمسجلات، وكواشف الإكمال لمرحلة الـ pipeline، حيث يوضح الشكل (4– أ) تنفيذ دالة منطقية تكافئ بوابة XNOR بمدخلين n=2 وخرج وحيد Z = X XNOR Y باستخدام بوابات العتبة كمثال عن كتلة المنطق، وتتطلب هذه الكتلة بنية مسجل NCL ثنائي البت وكاشف إكمال ثنائي البت موضحين في الشكل (4– ب). بشكل عام، يتكون مسجل NCL ثنائي البت من عدد 2n من بوابات العتبة TH22 ويُنفذ كاشف الإكمال ثنائي البت باستخدام من عدد n من بوابات OR ثنائية المدخل (أي TH12) وعنصر C ثنائي المدخل (أي THnn). يُستخدم كاشف الإكمال $CD_i$ في المرحلة $S_i$ لاستشعار ما إذا كان خرج المسجل $R_i$ هو DATA أو NULL حيث ينتقل خرج $CD_i$ من واحد منطقي إلى صفر منطقي عندما تأخذ جميع بتات المسجل $R_i$ القيمة NULL والعكس صحيح.

يتألف تدفق البيانات ضمن NCL pipeline من سلسلة من موجتي NULL و DATA متناوبتين. أي أنه توجد دائمًا جبهة موجة NULL/DATA بين موجتي DATA/NULL متتاليتين خلال تدفق البيانات. كما هو موضح في الشكل (3)، يتم توصيل معكوس خرج كاشف الإكمال $Ko_i$ في المرحلة $S_i$ بإشارة التحكم $Ki_{i-1}$ للمسجل $R_{i-1}$ في المرحلة السابقة $S_{i-1}$. عندما تمر موجة

البيانات أو موجة الـ NULL عبر الكتلة المنطقية $L_i$ ويتم حفظها بنجاح ضمن المسجل $R_i$، يصبح $Ko_i = 0$ و $Ki_{i-1} = 1$، مما يُمكّن المسجل $R_{i-1}$ في المرحلة السابقة من الاحتفاظ بقيم البيانات أو قيم NULL التالية.

على الرغم من أن تصميم الـ pipeline باستخدام المنطق NCL يتميز باستهلاك طاقة منخفض كونه لا يستهلك طاقة في الحالة غير النشطة إلا أنه لا يزال يعاني من تبديد طاقة التسرب الساكنة لذلك تم اقتراح عدد من التقنيات لتصميم بوابات على مستوى الترانزستور تتميز بالتحكم الدقيق بالطاقة بهدف تقليل طاقة التسرب الساكنة في الدارات غير المتزامنة. أحد هذه التقنيات هو NCL متعدد العتبات MTNCL ( Multi-threshold NCL ) [33]، حيث يستخدم MTNCL ترانزستورات CMOS متعددة العتبات لتحقيق بوابات عتبة تتحكم بالطاقة بشكل دقيق، وقد تم اقتراحه أولاً في [34].



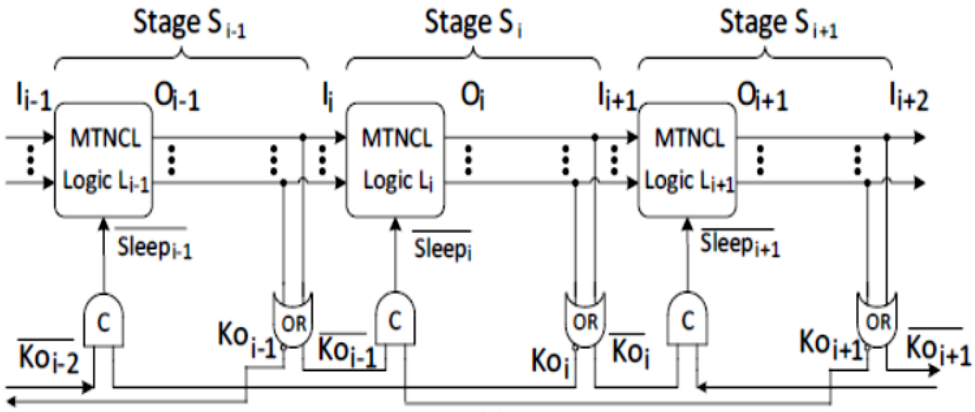

– ب –  – أ –

**الشكل(5): أ- البنية العامة لبوابة العتبة MTNCL – ب- بنية البوابة TH23 في المنطق MTNCL.**

يوضح الشكل (5- أ) البنية العامة لبوابات العتبة في المنطق MTNCL [33]، والتي تتكون من كتلتين وظيفيتين hold-0 و set-to-1، وترانزستورين سكونيين عاليي الجهد (Q1 و Q2) للتحكم الدقيق في استهلاك الطاقة بالإضافة إلى عاكس على الخرج مزود بترانزستور سحب للأسفل Q3. يظهر الشكل (5- ب) بوابة العتبة TH23 كمثال على هذه البوابات

تعمل بوابة العتبة MTNCL إما في الوضع النشط (مع إلغاء تفعيل إشارة التحكم Sleep) أو في وضع السكون (مع تفعيل إشارة التحكم Sleep). إذا كان الدخل الحالي هو DATA ويوجد صفر منطقي على مدخل أشارة التحكم $Sleep_i$، فيتم تشغيل الترانزستورين Q1 و Q2 في الشكل (5-أ)، وإيقاف تشغيل الترانزستور Q3 وتبدأ بوابات العتبة MTNCL بإعادة ضبط القيم على خرجها، وفي النهاية يصبح الخرج هو قيم البيانات الناتجة DATA. أما عند وجود قيمة NULL على الدخل وواحد منطقي على المدخل $Sleep_i$، فيتم إيقاف تشغيل الترانزستورين Q1 و Q2 مما يؤدي إلى وضع جميع بوابات العتبة بحالة خمول (بهدف الحد من استهلاك الطاقة)، وتشغيل الترانزستور Q3، وبالتالي يتم إجبار خرج جميع البوابات ليصبح صفر منطقي (أي يأخذ الخرج القيمة NULL).

استخدمت العديد من الداراسات المنطق MTNCL في تصميم الـ pipeline وتطويرها [32,31] مثل MTNCL pipeline التقليدي الذي يختلف عن الـ NCL pipeline بالتصميم الداخلي لبوابات العتبة والـ Register-Less NCL pipeline (RL-NCL) الذي تم خلاله التعديل على بنية الـ MTNCL pipeline التقليدي وإلغاء جميع المسجلات، وبالتالي تقليل استهلاك المساحة والطاقة.




**الشكل (6): بنية RL-NCL pipeline.**

يوضح الشكل (6) بنية RL–NCL pipeline المقترح في [31]، حيث تتألف مرحلة الـ pipeline المشار إليها بالرمز $S_i$ من ثلاث مكونات: 1) كتلة منطقية Li، مُنفذة باستخدام بوابات العتبة MTNCL، 2) كاشف إكمال $CD_i$ عبارة عن بوابة OR للكشف عما إذا كان خرج الكتلة المنطقية DATA أو NULL، 3) عنصر C، يُستخدم للتحكم في وضع التشغيل (بحالة نشطة أو سكون) للكتلة المنطقية Li فهو يولد إشارة السكون (Sleep) التي تستخدمها بوابات العتبة لإبقاء البوابة المعنية نشطة أو لوضعها بحالة سكون، وبالتالي تقليل استخدام بوابات MTNCL ضمن الكتلة المنطقية.

في [31] تمت المقارنة بين البنى الثلاثة المذكورة سابقاً وهي NCL pipeline و MTNCL pipeline و RL–NCL pipeline حيث استخدمت لتنفيذ جامع بطول 8 بت ويتطلب pipeline بخمس مراحل وأجريت هذه المحاكاة على مستوى الترانزستور باستخدام أداة المحاكاة HSPICE وتقنية 32 نانومتر.

**الجدول(2): مقارنة الأداء بين NCL pipeline وMTNCL pipeline و RL–NCL pipeline [31].**

| بنية الـ pipeline | | Conv. NCL | MTNCL | RL–NCL |
|---|---|---|---|---|
| استهلاك الطاقة (µw) | معدل دخل بيانات = 10 ميجاهرتز | 10.1 | 8.0 | 2.8 |
| | معدل دخل بيانات = 700 ميجاهرتز | 176.7 | 177.1 | 66.3 |
| استهلاك الموارد | عدد الترانزستورات | 6207 | 5750 | 3132 |
| | المساحة ($nm^2$) | 10415154 | 9226624 | 4765184 |

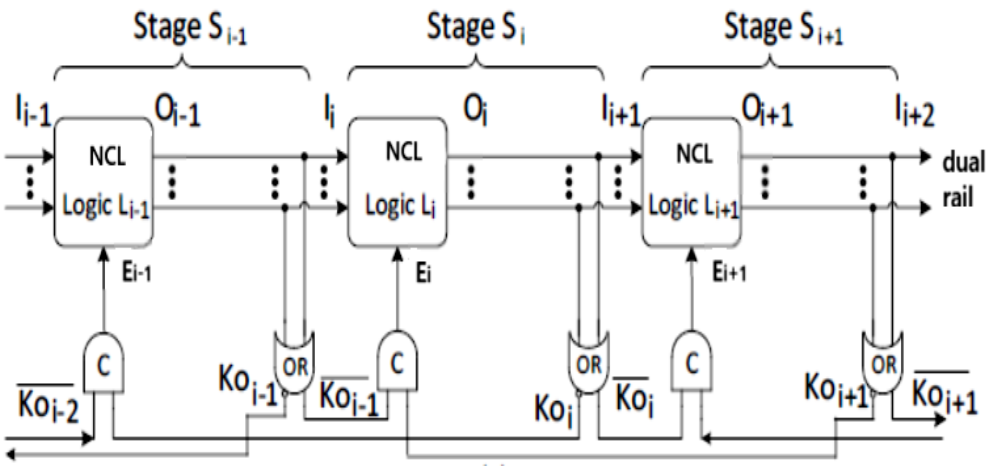
حقق RL–NCL pipeline وفقاً لنتائج المحاكاة المذكورة في الدراسة والتي تم تلخيصها في الجدول(2) استهلاكًا أقل للطاقة مقارنةً بـ NCL التقليدي سواءً عند معدلات دخل منخفضة أو عالية، وذلك بسبب أولاً التخلص من المسجلات، ثانياً استبدال كواشف الإكمال المعقدة ببوابات OR أبسط، وثالثاً التخفيف من تسرب الطاقة في الكتل المنطقية الخاملة عن طريق التحكم الدقيق

في استهلاك الطاقة. حيث يمكن لـ RL–NCL pipeline تقليل استهلاك الطاقة بنسبة 72.5% و62.5% عند معدل دخل بيانات يبلغ 10 ميجاهرتز و700 ميجاهرتز على التوالي مقارنةً بنظيره التقليدي NCL pipeline. في حين يوفر استهلاك مساحة بمقدار 45.8% وعدد ترانزستورات بمقدار 50.5%.

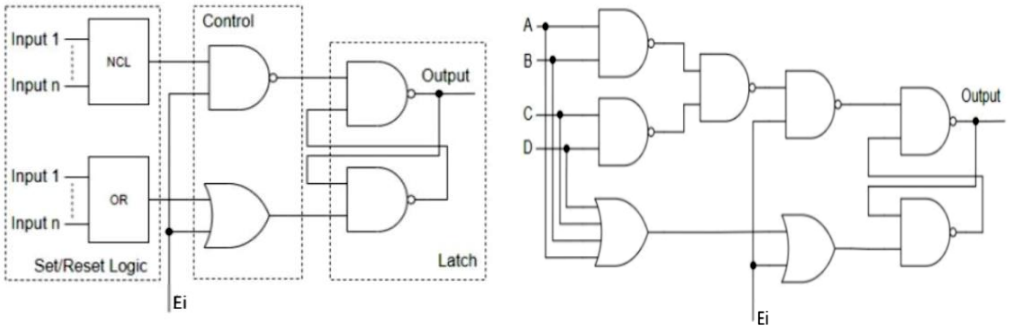
العيب الوحيد في بنية RL–NCL pipeline هو أن المشروع مخصص بالكامل تم تنفيذه على مستوى الترانزستور باستخدام بوابات المنطق MTNCL وبالتالي لا يمكن الاستفادة منه قي تنفيذ التصاميم على مستوى البوابة التي تستخدم البوابات القياسية للمنطق NCL كما هي الحال في تصميمنا لموجه الشبكات ضمن الرقاقة لذلك بهدف تحسين الأداء وتقليل استهلاك الموارد في الشبكات ضمن الرقاقة القائمة على المنطق NCL ومن خلال الاستفادة من مزايا RL–NCL pipeline تم في هذا البحث اقتراح طريقة جديدة لتنفيذ بوابات NCL تغني عن حاجة RL–NCL pipeline لبوابات MTNCL وتجعله قابلاً للتنفيذ على FPGAs أو باستخدام خلايا المنطق NCL القياسية.

### 6. بنية pipeline المقترحة:

يوضح الشكل (7) البنية المقترحة التي تشبه إلى حد كبير بنية RL–NCL pipeline ولكن باعتماد طريقة جديدة لتنفيذ بوابات NCL ضمن الكتلة المنطقية، بحيث يتم تحقيق نفس مزايا البنية السابقة باستخدام بوابات NCL القياسية بدلاً من استخدام بوابات المنطق MTNCL التي تم تحقيقها على مستوى الترانزستور وكانت مخصصة بالكامل.




**الشكل (7): بنية pipeline المقترحة.**




– ب –  –أ –

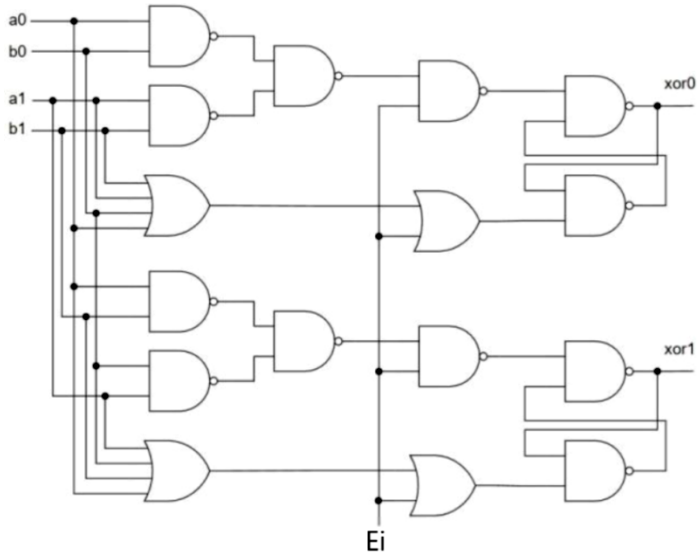
**الشكل(8): طريقة التنفيذ المقترحة على مستوى البوابة: أ– البنية العامة – ب– بنية البوابة THxor0 كمثال توضيحي.**

يوضح الشكل (8– أ) بنية الدارة المستخدمة لتنفيذ بوابات NCL ضمن الكتلة المنطقية للبنية المقترحة، الهدف من هذه الدارة هو محاكاة سلوك بوابات المنطق MTNCL باستخدام بوابات NCL القياسية، وهي تتألف من ثلاثة أجزاء الماسك ، وكتلة التحكم، ومنطق الضبط/إعادة الضبط. الماسك هو ماسك RS قياسي يستخدم بوابات NAND، في حين تستخدم كتلة التحكم لتفعيل

الماسك RS فعندما يكون على مدخل التفعيل $E_i$ واحد منطقي، يجب أن تُمكّن الماسك من ضبط خرجه ومنع حالة إعادة الضبط، أما عندما تكون إشارة $E_i$ صفر منطقي وجميع المداخل الأساسية بحالة NULL، يتم إعادة ضبط الماسك ومنع حالة الضبط. وبالتالي، تستخدم كتلة التحكم بوابة NAND لإشارة الضبط وبوابة OR لإشارة إعادة ضبط الماسك. تعتمد بنية الضبط وإعادة الضبط على دارة NCL المراد تنفيذها، فمن أجل إعادة الضبط يتم استخدام بوابة OR واحدة عدد مداخلها يساوي عدد مداخل دارة NCL المطلوبة ضمن جزء الضبط وبالتالي يكون خرج بوابة OR صفر منطقي فقط عندما يكون على جميع المداخل صفر منطقي (حالة NULL) وهذا بدوره يضمن أيضاً إلغاء حالة عدم الاستقرار من خلال إعادة ضبط الخرج فقط عندما تكون جميع المداخل صفر منطقي بينما يتمثل جزء الضبط بدالة NCL التي تعبر عن وظيفة الكتلة المنطقية.

على سبيل المثال، بفرض الكتلة المنطقية عبارة عن بوابة THxor0. يوضح الشكل (8– ب) بنية هذه الكتلة باستخدام الطريقة المقترحة، حيث تم استخدام ثلاث بوابات NAND لتنفيذ الدالة المنطقية AB+CD المكافئة للبوابة THxor0 والتي تعبر عن جزء الضبط ضمن الكتلة المنطقية أما بالنسبة لإعادة الضبط فهو عبارة عن بوابة OR بأربع مداخل تبعاً لعدد المداخل الأساسية A وB وC وD للبوابة THxor0. عندما يكون على مدخل التفعيل $E_i$ واحد منطقي، يُسمح لخرج الضبط الذي يعبر عن دالة البوابة بالوصول إلى خرج الكتلة المنطقية، وبمجرد ضبط الخرج، لن يُعاد ضبطه مرة أخرى إلا عندما يكون على المدخل $E_i$ صفر منطقي وعلى جميع مداخل البوابة صفر منطقي أيضاً. يبين الشكل(9) البنية الموسعة لبوابة XOR بمدخلين وترميز ثنائي الأسلاك باستخدام طريقة التنفيذ المقترحة.

وهنا يجب التنويه إلى أن جميع الدارات المُنفذة تم بناءها أولاً باستخدام بوابات المنطق البولياني كما هو مبين في الشكلين (9,8) ثم تم تحويلها إلى شبكة من بوابات NCL المكافئة باستخدام أداة خاصة بالمنطق NCL وسوف نتطرق لذلك بالتفصيل ضمن قسم المحاكاة والنتائج خلال هذا البحث.




**الشكل (9): بنية بوابة XOR بمدخلين وترميز ثنائي الأسلاك باستخدام طريقة التنفيذ المقترحة.**

يتم توليد أشارة التفعيل $E_i$ باستخدام عنصر C بمدخلين $Ko_{i-1}$ و $Ko_{i+1}$ أي أنه لا يمكن للكتلة المنطقية Li ضمن المرحلة $S_i$ بدء عملية معالجة بيانات جديدة $D_k$ أو إبطال البيانات السابقة وتوليد قيمة NULL على الخرج $N_k$ عند $I_{i+1}$ حتى تصل قيمة NULL السابقة $N_{k-1}$ أو قيمة البيانات السابقة $D_{k-1}$ بأمان إلى المدخل $I_{i+2}$ في المرحلة $S_{i+2}$. يمنع هذا التقييد قيم DATA/NULL الحالية $D_k/N_k$ من تجاوز قيم NULL/DATA السابقة $N_{k-1}/D_{k-1}$.

تبدأ جميع بوابات العتبة ضمن الكتلة المنطقية معالجة قيم البيانات الجديدة أو إبطال القيم السابقة وتوليد قيم NULL على خرجها في الوقت نفسه، بحيث يأخذ بت الخرج على المسار الحرج للكتلة المنطقية قيمة DATA أو NULL بعد أن تاخذ جميع بتات الخرج الأخرى قيم DATA أو NULL. لذلك، يمكن لمرحلة الـ pipeline المقترحة استخدام بوابة OR بمدخلين فقط لتحل محل كاشف الإكمال للكشف عما إذا كان خرج الكتلة المنطقية DATA أو NULL حيث يتصل مدخليها بزوج الأسلاك المرتبط ببت الخرج على المسار الحرج.

**يمكننا توضيح آلية عمل الـ pipeline المقترحة من خلال الخطوات التالية:**

بدايةً: بفرض أن كتلة المنطق Li دخلت للتو في وضعية NULL (أي $E_i = 0$)، مما أدى إلى أن يصبح الخرج NULL (أي أن رمز NULL المُشار إليه بـ $N_{k-1}$، موجود الآن عند $O_i$).

1) عندما تصل بيانات جديدة $D_k$ إلى الدخل $I_i$ للمرحلة $S_i$ يصبح معكوس $Ko_{i-1}$ يساوي الواحد منطقي ويقوم جزء الضبط ضمن الكتلة المنطقية $L_i$ بمعالجة هذه البيانات ولكن خرج المرحلة $O_i$ سيبقى في وضع NULL حتى يصبح المدخل الثاني للعنصر C $Ko_{i+1}$ يساوي الواحد منطقي أيضاً.
2) بعد أن يصبح $Ko_{i+1}$ مساوياً للواحد منطقي (أي أن كتلة المنطق التالية $L_{i+1}$ قد دخلت في وضع NULL ووصل رمز NULL السابق $N_{k-1}$ بنجاح إلى المدخل $I_{i+2}$)، تتغير إشارة التفعيل $E_i$ إلى الواحد منطقي مما يسمح لخرج جزء الضبط ضمن الكتلة المنطقية بالوصول إلى خرج المرحلة $O_i$.
3) تصل رزمة البيانات الجديدة $D_k$ إلى الدخل $I_{i+1}$ للمرحلة التالية $S_{i+1}$ في حين يصبح على الدخل $I_i$ للمرحلة $S_i$ قيم NULL مجدداً $N_k$، وبالتالي يتغير معكوس $Ko_{i-1}$ إلى الصفر منطقي.
4) بعد أن يُصبح $Ko_{i+1}$ مساوٍ للصفر منطقي (أي أن قيمة البيانات السابقة $D_k$ أصبحت على خرج الكتلة المنطقية $L_{i+1}$ ووصلت إلى 2+Ii)، تدخل الكتلة المنطقية Li بوضع الـ NULL (Ei = 0 ) من جديد.
5) في النهاية، تُكمل كتلة المنطق $L_i$ عملية إبطال البيانات السابقة ويصبح على خرجها قيم NULL ($N_k$) التي تصل إلى الدخل $I_{i+1}$ للمرحلة التالية $S_{i+1}$.
6) ويتم تكرار الخطوات السابقة مع رزم البيانات الجديدة.

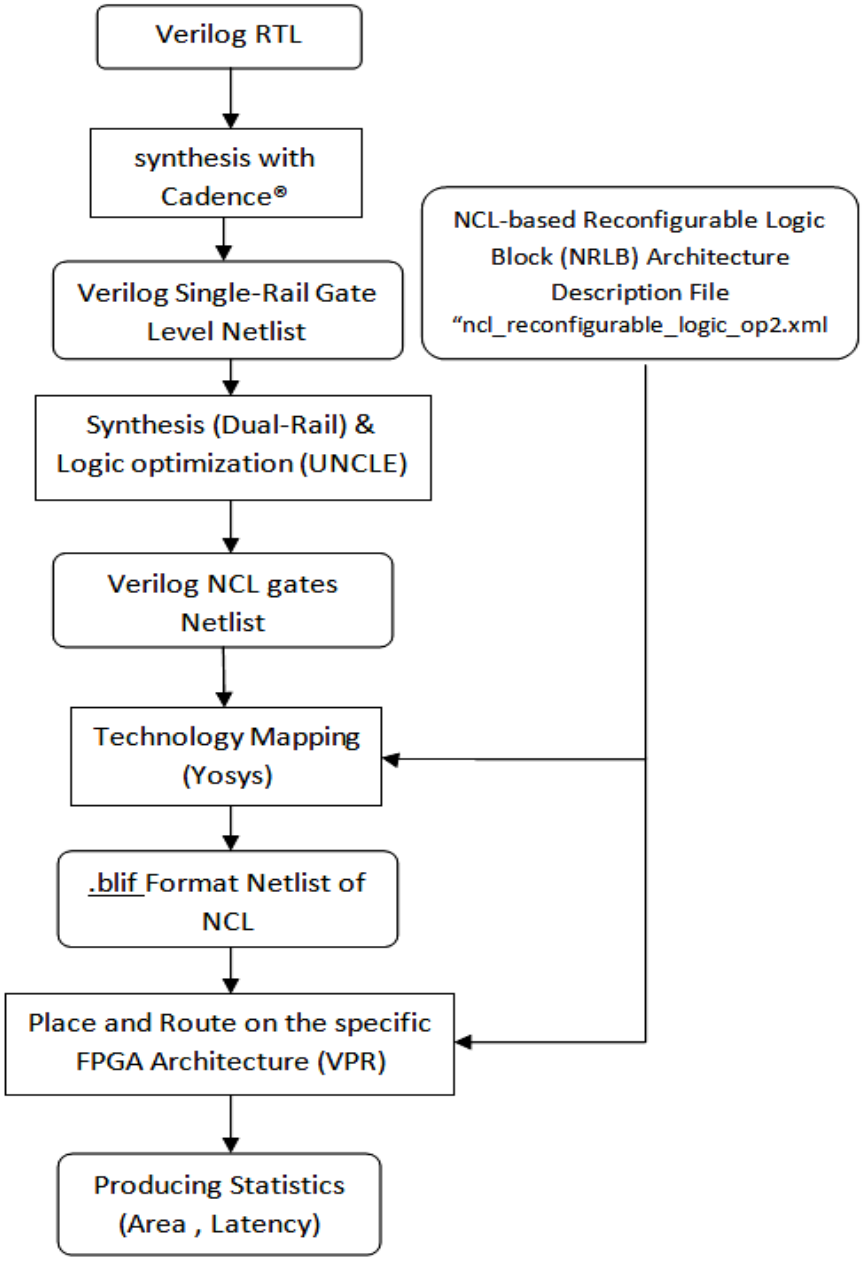
## 7. المحاكاة والنتائج:

### 1.7. تدفق التصميم والمحاكاة:

للتحقق من فعالية الطريقة المقترحة لتنفيذ بوابات NCL ضمن بنية pipeline الجديدة ودورها في تحسين أداء الشبكات ضمن الرقاقة وتقليل استهلاك الموارد، قمنا بتنفيذ موجه الشبكة باستخدام البنية المقترحة وبنية NCL pipeline التقليدية لمقارنة الأداء، وذلك من خلال الاستعانة بتدفق التصميم VTR(Verilog-to-Routing) [35] كونه يدعم التصاميم غير المتزامنة ويوفر

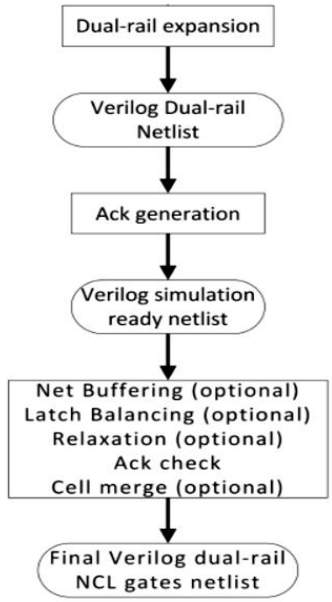
إمكانية تحقيقها على بنى قابلة لإعادة البرمجة مضمنة أو يمكن للمصمم استخدام بنية قابلة لإعادة البرمجة خارجية بإضافة ملف المواصفات الخاص بها إلى تدفق التصميم كما هو الحال في هذا البحث.

استنادًا إلى تدفق تصميم CAD (Computer Aided–Design) المُخصّص لـ VTR المبين في الشكل(10) تم بناء تصميم Verilog RTL للوحدات الفرعية لموجه NoC الموصوف في [39] باستخدام بوابات منطقية بوليانية أحادية الأسلاك من خلال أداة المحاكاة ®Cadence في المرحلة الأولى. بعد ذلك، استُخدمت أداة UNCLE( Unified NCL Environment ) [36]، المُبينة في الشكل (11) لتنفيذ البنية غير المتزامنة بتوليف شبكة البوابات المنطقية أحادية الأسلاك من المرحلة السابقة إلى شبكة ثنائية الأسلاك باستخدام بوابات العتبة NCL، ثم تم استخدام الأداة Yosys [37] لتنفيذ خطوة تعيين شبكة بوابات NCL على الرقاقة باستخدام البنية القابلة لإعادة البرمجة NRLB (NCL based Reconfigurable Logic Block ) [38] من خلال تضمين ملف المواصفات الخاص بها، وفي الخرج نحصل على ملف BLIF لبنية الدارة على مستوى البوابة. في المرحلة ما قبل الأخيرة، تُنفذ الأداة VPR ( Versatile Place and Route ) [35] عملية تحقيق الدارة وتوصيلها على بنية NRLB. وأخيرًا، استُخرجت إحصائيات المساحة والتوقيت بواسطة VPR لتحليلها لاحقًا ومقارنتها للحصول على النتائج [35].




**الشكل(10): تدفق التصميم VTR.**

استنادًا إلى تدفق تصميم CAD (Computer Aided–Design) المُخصّص لـ VTR المبين في الشكل(10) تم بناء تصميم Verilog RTL للوحدات الفرعية لموجه NoC الموصوف في [39] باستخدام بوابات منطقية بوليانية أحادية الأسلاك من خلال أداة المحاكاة ®Cadence في المرحلة الأولى. بعد ذلك، استُخدمت أداة UNCLE( Unified NCL Environment ) [36]، المُبينة في الشكل (11) لتنفيذ البنية غير المتزامنة بتوليف شبكة البوابات المنطقية أحادية الأسلاك من المرحلة السابقة إلى شبكة ثنائية الأسلاك باستخدام بوابات العتبة NCL، ثم تم استخدام الأداة Yosys [37] لتنفيذ خطوة تعيين شبكة بوابات NCL على الرقاقة باستخدام البنية القابلة لإعادة البرمجة NRLB (NCL based Reconfigurable Logic Block ) [38] من خلال تضمين ملف المواصفات الخاص بها، وفي الخرج نحصل على ملف BLIF لبنية الدارة على مستوى البوابة. في المرحلة ما قبل الأخيرة، تُنفذ الأداة VPR ( Versatile Place and Route ) [35] عملية تحقيق الدارة وتوصيلها على بنية NRLB. وأخيرًا، استُخرجت إحصائيات المساحة والتوقيت بواسطة VPR لتحليلها لاحقًا ومقارنتها للحصول على النتائج [35].

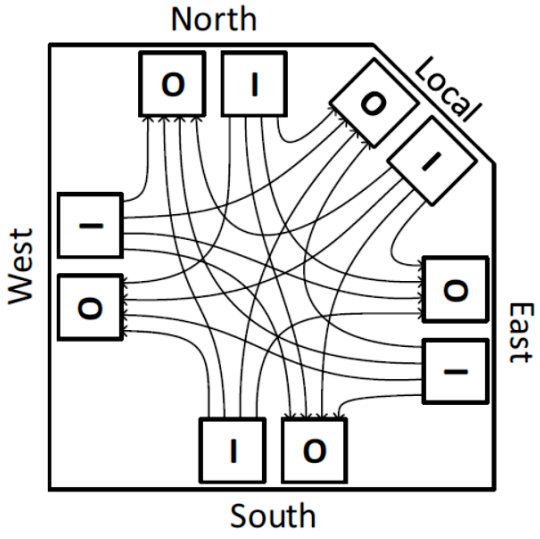



**الشكل(11): تدفق تصميم أداة المحاكاة UNCLE.**

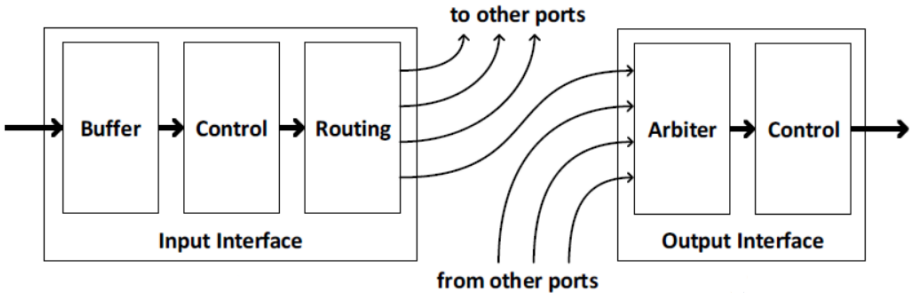
يبين الشكل (11) تدفق التصميم لأداة UNCLE [36] التي تم تنفيذها باستخدام نصوص Python® وتستخدم هذه الأداة مكتبة بوابات NCL قياسية في عملية التعيين لتحويل البوابات المنطقية البوليانية أحادية الأسلاك إلى بوابات عتبة NCL ثنائية الأسلاك، وتعطي في خرجها شبكة Verilog NCL ثنائية الأسلاك النهائية بعد إضافة شبكة المصافحة مع بعض خطوات التحسين الاختيارية.

استخدمت الكتلة المنطقية القابلة لإعادة البرمجة القائمة على المنطق NCL (NRLB) [38] كمكون للنظام القابل لإعادة البرمجة على الرقاقة، حيث تم تطويرها وتقييمها باستخدام Cadence® Virtuoso® Layout Suite مع تقنية أشباه الموصلات 45 FDSOI CMOS نانومتر. واستخدم محاكي Cadence® Spectre® AMS لمحاكاة بوابات NCL القابلة لإعادة البرمجة واستخراج نتائج تأخير البوابة التي تم توظيفها لاحقًا ضمن ملف وصف البنية المطلوب لتدفق VTR المستخدم لتقييم أداء موجه NoC.

يبين الشكلان (12) و (13) البنية المعتمدة لموجه NOC [39] وبنية أحد منافذه على التوالي وكما هو موضح في الشكل (14) تم تجزئة بنية الموجه إلى ستة مراحل pipeline تمر ضمنها رزمة البيانات لتصل إلى خرج الموجه بأمان ويتضمن الجدول (3) البارامترات الأساسية بهذا الموجه التي تم اعتمادها في عملية المحاكاة.




الشكل(12): البنية المعتمدة للموجه NOC.




الشكل(13): بنية المنفذ ضمن موجه NOC.

| Stage 1 | Stage 2 | Stage 3 | Stage 4 | Stage 5 | Stage 6 |
|---|---|---|---|---|---|
| Buffer | Input Control | Routing | Switch Traversal | Arbiter | Output Control |

الشكل(14): مراحل الـ pipeline ضمن الموجه.

**الجدول(3): البارامترات الأساسية لموجه NOC المعتمدة ضمن المحاكاة [39].**

| Router Parameters | |
|---|---|
| Flit Data Width (bits) | 32 |
| Number of Ports | 5 |
| Buffer Depth on each port (Flits) | 32 |
| Router pipeline stages | 6 |
| Routing Algorithm | Dimension-order (XY) |
| Arbiter | Round-robin policy |

### 2.7. النتائج:

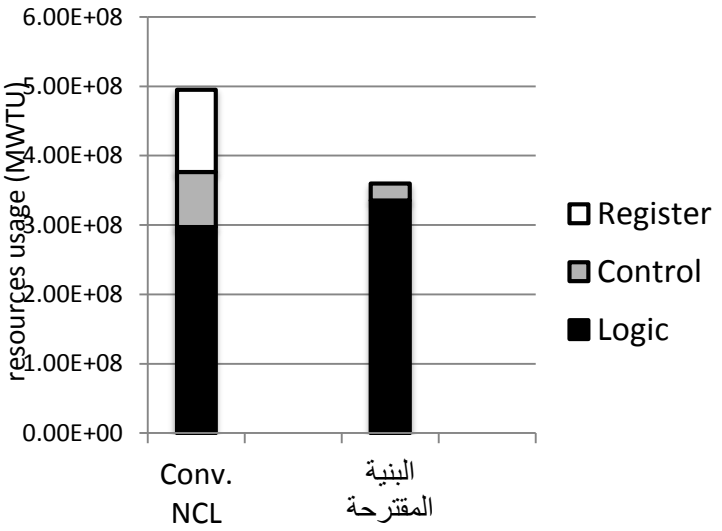
تم الاستناد في النتائج إلى التقارير الزمنية وتقارير VPR الناتجة عن تدفق التصميم السابق لمقارنة أداء الموجه باستخدام NCL pipeline التقليدي والـ pipeline المقترح، استخدمنا ثلاث بارامترات: (1) زمن التأخير (بالنانوثانية)، وهو الوقت اللازم لرزمة البيانات الجديدة لعبور جميع مراحل الـ pipeline والوصول إلى خرج الموجه. (2) عدد الترانزستورات ذات العرض الأدنى MWTU ( minimum width transistor unit ) وهي الوحدة المستخدمة لقياس الموارد المستهلكة في أداة التصميم بمساعدة الحاسب الأكاديمية مفتوحة المصدر VPR ( العرض الأدنى للترانزستور في تقنية أشباه الموصلات 45nm CMOS يساوي 120 nm وبالتالي مساحة الترانزستور ذو العرض الأدنى $29.952 \times 10^{-3}$ $um^2$ ). (3) عدد وحدات NRLB المستخدمة.

كما هو مبين في الجدول (4) فإن بنية pipeline المقترحة تحقق انخفاضاً باستهلاك موجه الشبكات ضمن الرقاقة للموارد إن كان بعدد وحدات NRLB القابلة للبرمجة المستخدمة أو بعدد الترانزستورات المقاسة بـ MWTU بنسبة 27.32% بشكل وسطي بينما يقلل زمن تأخير الموجه بنسبة 14.1% مقارنة بـ NCL pipeline التقليدي والسبب يعود إلى الطريقة المقترحة في تنفيذ بوابات NCL ضمن الكتلة المنطقية لمرحلة الـ pipeline والاستفادة من بنية RL-NCL pipeline حيث وفرت هذه الطريقة تحقيق تصميم RL-NCL pipeline على مستوى البوابة

باستخدام مكتبة بوابات المنطق NCL القياسية دون الحاجة لبوابات المنطق MTNCL بالإضافة إلى التخلي عن مسجلات الـ pipeline واستبدال دارة كاشف الإكمال المعقدة ببوابات OR بسيطة حيث يبين الشكل (15) مخطط بياني للموارد المستهلكة من قبل كل من الكتلة المنطقية ووحدة التحكم ( كواشف الإكمال وعناصر C وغيرها ) والمسجلات ضمن الموجه باستخدام NCL pipeline التقليدي والـ pipeline المقترح ففي التقنية التقليدية تستهلك مسجلات الـ pipeline ودارة التحكم نحو 40% من الموارد المستخدمة ضمن الموجه في حين يبلغ استهلاك وحدة التحكم اقل من 10% من أجمالي استهلاك الموجه للموارد عند تنفيذ الـ pipeline بالطريقة المقترحة وهنا لا بد من الإشارة إلى أن هذه الطريقة تؤدي إلى زيادة بالموارد المستهلكة من قبل الكتلة المنطقية في البنية الجديدة وهي زيادة مقبولة طالما أنها تحافظ على مزايا RL–NCL pipeline وتقلل التأخير والموارد المستهلكة في الشبكات ضمن الرقاقة والذي بدوره يوفر استهلاك الطاقة أيضاً.

**الجدول(4): مقارنة النتائج بين الـ pipeline المقترحة و NCL pipeline التقليدي.**

| بنية الـ pipeline | | Conv. NCL | البنية المقترحة | نسبة التحسن المئوية |
|---|---|---|---|---|
| **الموارد المستهلكة ضمن الموجه** | **MWTU** | 4.95125e+08 | 3.598568e+08 | وسطياً 27,32% |
| | **عدد NRLBs** | 41624 | 30253 | |
| **تأخير الموجه(ns)** | | 1389.79 | 1193.829 | 14,1% |




**الشكل(15): مخطط بياني للموارد المستهلكة ضمن الموجه باستخدام البنية المقترحة والبنية التقليدية.**

## 8. الخاتمة:

تلعب تقنية الـ pipeline في الشبكات ضمن الرقاقة القائمة على المنطق NCL دوراً أساسياً في تنظيم تدفق البيانات ضمن الشبكة ولكنها تستهلك جزء كبير من الموارد لذلك تم خلال هذا البحث اقتراح بنية جديدة للـ pipeline بهدف تحسين أداء الشبكة وتقليل استهلاك الموارد انطلاقاً من البنية RL–NCL pipeline حيث تم التعديل على هذه البنية بحيث تصبح قابلة للتحقيق باستخدام بنى NCL القابلة لإعادة البرمجة من خلال اقتراح طريقة جديدة لتنفيذ بوابات NCL ضمن الكتلة المنطقية للـ pipeline بحيث تقدم سلوك مشابه لسلوك بوابات المنطق MTNCL المستخدمة ضمن البنية الأساسية للـ RL–NCL pipeline وبالتالي الاستفادة من مزايا هذه التقنية واستخدامها في تصميم بنية موجه الشبكات ضمن الرقاقة غير المتزامنة القائمة على المنطق NCL حيث أثبتت نتائج المحاكاة انخفاض استهلاك الموارد من قبل موجه الشبكات ضمن الرقاقة الذي يستخدم

بنية pipeline المقترحة بنسبة % 27.32 مقارنة بالبنية المُنفذة باستخدام NCL pipeline التقليدية بالإضافة إلى تقليل زمن تأخير الموجه بنسبة % 14.1 .

## 9. المراجع:

**[1]** B. H. Calhoun, Y. Cao, X. Li, K. Mai, L. T. Pileggi, R. A. Rutenbar, and K. L. Shepard, "Digital circuit design challenges and opportunities in the era of nanoscale cmos", Proceedings of the IEEE, vol. 96, no. 2, pp. 343–365, 2008.

**[2]** J. Sparsø, "Introduction to Asynchronous Circuit Design". DTU Compute, Technical University of Denmark, 2020.

**[3]** K. M. Fant and S. A. Brandt, "NULL convention logic: A complete and consistent logic for asynchronous digital circuit synthesis", in Proc. Int. Conf. Appl. Specific Syst. Archit. Process., pp. 261–273, 1996.

**[4]** R. Parikh, R. Das, and V. Bertacco, "Power-aware NoCs through routing and topology reconfiguration", Proceedings of the ACM/IEEE Design Automation Conference (DAC), pp.1–6, 2014.

**[5]** S. M. Nowick and M. Singh, "High-performance asynchronous pipelines: An overview" , Ieee design & test of computers, vol. 28, no. 5, pp. 8–22, 2011.

**[6]** S.M. Nowick and M. Singh, "Asynchronous design-Part 1: overview and recent advances", IEEE Des. Test 32 (3), pp. 5–18, 2015.

**[7]** A. Lines, "The vortex: a superscalar asynchronous processor", Proceedings of the 13th IEEE International Symposium on Asynchronous Circuits and Systems, ASYNC, pp. 39–48, 2007.

**[8]** J. Dama and A. Lines, "GHz asynchronous SRAM in 65 nm", Proceedings of the 15th IEEE Symposium on Asynchronous Circuits and Systems, ASYNC, pp. 85–94, 2009.

**[9]** J. Tse and A. Lines, "NanoMesh: an asynchronous kilo-core system-on-chip", Proceedings of the IEEE 19th International Symposium , Asynchronous Circuits and Systems (ASYNC) May. 2013, pp. 40-49, 2013.

**[10]** M. Singh and S. M. Nowick, "The design of high performance dynamic asynchronous pipelines: Lookahead style", IEEE Trans. VLSI Syst., vol. 15, no. 11, pp. 1256-1269, 2007.

**[11]** M. Singh and S. M. Nowick, "The design of high performance dynamic asynchronous pipelines: High capacity style", IEEE Trans. VLSI Syst., vol. 15, no. 11, pp.1270-1283, 2007.

**[12]** D. L. Oliveira, G. C. Duarte, N. N. M. Cardoso and G. C. Batista, "Implementation of Asynchronous Pipelines with QDI Template onto FPGAs Using Commercial Tools", *2020 33rd Symposium on Integrated Circuits and Systems Design (SBCCI)*, Campinas, Brazil, pp. 1-5, 2020.

**[13]** D. L. Oliveira, G. C. Duarte and G. C. Batista, "A New QDI Asynchronous Pipeline with Two-Phase Delay-Insensitive Global Communication", 2021 IEEE 12th Latin America Symposium on Circuits and System (LASCAS), Arequipa, Peru, pp. 1-4, 2021.

**[14]** G. C. Duarte, D. L. Oliveira and G. C. Batista, "Design of Asynchronous Pipelines with QDI Template Using Commercial FPGA", 2022 IEEE 13th Latin America Symposium on Circuits and System (LASCAS), Puerto Varas, Chile, pp. 1-4, 2022.

**[15]** M. Davies, N. Srinivasa, T. H. Lin, G. Chinya, Y. Q. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, et al., "Loihi: A neuromorphic manycore processor with on-chip learning", IEEE Micro, vol. 38, no. 1, pp. 82-99, 2018.

**[16]** A. Mardari, Z. Jelˇcicov´a, and J. Sparsø, "Design and FPGAimplementation
of asynchronous circuits using two-phase handshaking", in Proc. 2019 25th Int. Symp. Asynchronous Circuits and Systems, Hirosaki, Japan, pp. 9–18, 2019.

**[17]** A. Ghafoor, M. W. Mughal and A. A. Khan, "An FPGA Compliant Single-Rail Encoded Asynchronous Pipeline", IEEE Access, vol. 9, pp. 160186-160194, 2021.

**[18]** S. Semba and H. Saito, "RTL Conversion Method From Pipelined Synchronous RTL Models Into Asynchronous Ones", IEEE Access, vol. 10, pp. 28949-28964, 2022.

**[19]** R.K. Kavitha, A. Khajamastan and Y.R. Akhilesh et al, "High-performance asynchronous pipeline using embedded delay element", Microprocessors and Microsystems, vol 73, pp. 102955-102962, 2020.

**[20]** Yu Zhou," Investigation of Asynchronous Pipeline Circuits Based on Bundled-Data Encoding: Implementation Styles, Behavioral Modeling, and Timing Analysis", TSINGHUA SCIENCE AND TECHNOLOGY, vol 27, Numbe r 3, pp. 559-580, 2022.

**[21]** P. A. Beerel, R. O. Ozdag, and M. Ferretti, "A Designer's Guide to Asynchronous VLSI", Cambridge, MA, USA :Cambridge University Press, 2010.

**[22]** M. Roncken, I. Sutherland, C. Chen, Y. Hei, W. Hunt, C. Chau, S. M. Gilla, H. Park, X. Y. Song, A. P. He, et al., "How to think about self-

timed systems", Proc. 51st Asilomar Conf. Signals, Systems, and Computers, Pacific Grove, CA, USA, pp. 1597–1604, 2017.

**[23]** M. Ferretti and P. A. Beerel, "High performance asynchronous design using single-track full-buffer standard cells", IEEE J. Solid-St. Circ., vol. 41, no. 6, pp. 1444–1454,2006.

**[24]** D. Edwards, W. Toms, S. Temple, L. Plana, J. Garside, and S. Furber, "The story of the amulet: A brief history of asynchronous events in Manchester", in This Asynchronous World, Essays dedicated to Alex Yakovlev on the occasion of his 60th birthday. 2nd ed. Newcastle University, Newcastle, UK, pp. 120–130, 2017.

**[25]** Sravani, K., Rao, R. "Novel Asynchronous Pipeline Architectures for High-Throughput Applications". Arab J Sci Eng 45, pp. 6625–6638, 2020.

**[26]** D. A. Edwards and W. B. Toms, "The status of asynchronous design in industry", Information Society Technologies (IST) Programme, Tech. Rep. IST-1999-29119, Jun. 2004.

**[27]** F. A. Parsan, W. K. Al-Assadi, and S. C. Smith, "Gate mapping automation for asynchronous NULL convention logic circuits", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 1, pp. 99–112, Jan. 2014.

**[28]** F. A. Parsan, S. C. Smith, and W. K. Al-Assadi, "Design for testability of sleep convention logic", IEEE Trans. Very Large Scale Integ. (VLSI) Syst., vol. 24, no. 2, pp. 743–753, Feb. 2016.

**[29]** Q. Ou, F. Luo, S. Li, and L. Chen, "Circuit level defines against fault attacks in pipelined NCL circuits", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 9, pp. 1903–1913, Sep. 2015.

**[30]** S. Smith and J. Di, "Designing Asynchronous Circuits using NULL Convention Logic (NCL)",2009.

**[31]** M. Chang, P. Yang, and Z. Pan, "Register-less null convention logic", IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 64, no. 3, pp. 314–318, 2017.

**[32]** M. -C. Chang, M. -H. Hsieh and P. -H. Yang, "Low-Power Asynchronous NCL Pipelines With Fine-Grain Power Gating and Early Sleep," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 61, no. 12, pp. 957-961, Dec. 2014.

**[33]** L. Zhou, R. Parameswaran, F. A. Parsan, S. C. Smith, and J. Di, "Multi-threshold null convention logic (mtncl): An ultra-low power asynchronous circuit design methodology", Journal of Low Power Electronics and Applications, vol. 5, no. 2, pp. 81–100, 2015.

**[34]** A. D. Bailey, J. Di, S. C. Smith, and H. A. Mantooth, "Ultra-low power delay-insensitive circuit design", in 2008 51st Midwest Symposium on Circuits and Systems, pp. 503–506, 2008.

**[35]** K.E. Murray et al., "VTR 8: High-performance CAD and Customizable FPGA Architecture Modelling", ACM Transactions on Reconfigurable Technology and Systems (TRETS), vol. 13, no. 9, p 1–55, 2020.

**[36]** R. B. Reese and R. A. TAYLOR. "Uncle (Unified NCL Environment)" [Online].Available:http://my.ece.msstate.edu/faculty/reese/uncle/UNCLE.pdf

**[37]** C. Wolf. "Yosys Open SYnthesis Suite". Available: http://www.clifford.at/yosys/

**[38]** J. Yu and P. Beckett, "A dual-rail LUT for reconfigurable logic using null convention logic". In Proceedings of the 24th edition of the great lakes symposium on VLSI (GLSVLSI'14). ACM, NY, USA, pp. 261-266, 2014.

**[39]** M. Moreira et al., "The YeAH! NoC Router", Faculty of Informatics,PUCRS, Tech. Rep. 083, Dec. 2014.

# دراسة تقنيات التوجيه في الشبكات المعرفة برمجياً باستخدام التعلم الآلي

المهندس أحمد محيو الشيخ الدكتور المهندس علي الحاتم

**الملخص**

يعمل توجيه التدفق على تحقيق تحسينات دقيقة في أداء الشبكة من خلال توجيه تدفقات حزم البيانات المختلفة عبر مسارات شبكية متنوعة. حيث يوفر التحكم المركزي في الشبكات المعرفة البرمجية (Software-Defined Network) SDN إطاراً لتحسينات مركزية في الشبكة، مثل توجيه التدفق الأمثل، إن توجيه تدفق يتكيف مع تغيرات الأحمال يتطلب نماذج متطورة. نهدف من خلال هذه الدراسة إلى اتباع نهج يعتمد على التعلم المعزز، حيث ينتج عن التوجيه المباشر للتدفق فضاء من الثنائية (حالة–إجراء) يزداد مع زيادة عدد التدفقات الجارية، مما يؤدي إلى ظهور مشكلة محتملة في قابلية التوسع. نشير إلى أن هذه المشكلة تنشأ بشكل أساسي من المساحة المطلوبة لتخزين جدول Q الذي يتضمن الثنائيات الخاصة بالتدفق، بالإضافة إلى الوقت اللازم للتعلم حتى الوصول إلى التقارب والذي يتطلب استكشاف هذا الفضاء الكبير من الحالات والأفعال. من جانب أخر، فإنّ هناك عدد قليل من الحسابات المطلوبة لإيجاد الحالات (ويتطلب فقط تحديث قيم Q في كل خطوة). تم حل مشكلة قابلية التوسع من خلال استخدام عدة متحكمات SDN Controllers بحيث يكون لكل متحكم عدد أقل من العقد والتدفقات لإدارتها، وبالتالي جدول Q أصغر وفضاء حالة–إجراء أصغر. يقوم كل متحكم بتحسين توجيه التدفق ضمن شبكة صغيرة ومعقولة باستخدام التعلم المعزز، حيث يعمل كل نظام مستقل (Autonomous System) AS بتحسين التوجيه داخل المجال الخاص به وقد أظهرت النتائج التجريبية أن هذا النهج يُحقّق تحسينات جوهرية في الأداء: حيث بلغ متوسط زمن التقارب للمتحكمات المتعددة 10 خطوات فقط، مقارنةً بـ 40.5 خطوة في حالة المتحكم

المركزي الوحيد. كما انخفض متوسط زمن الاستجابة من 31.30 مللي ثانية (في النظام الأحادي) إلى 23.14 و23.99 مللي ثانية في أنظمة المتحكمات المتعددة. هذه النتائج تؤكد أن توزيع مسؤوليات التحكم يقلل بشكل فعّال من تعقيد فضاء التعلم، ويسرع من عملية التقارب، ويحسّن كفاءة توجيه التدفق، مما يجعل الحل المقترح قابلاً للتطبيق في الشبكات الكبيرة والديناميكية.

الكلمات المفتاحية: الشبكات المعَرفة برمجياً، توجيه التدفق، التعلم المعزز، التوسع.

## *Studying Routing Techniques in SDN Software-Defined Networking Using Machine Learning*

ENG.Ahmad Alshiekh Dr.Ali **Alhatem**

### Abstract

Flow routing enables fine-grained improvements in network performance by directing different data packet flows through diverse network paths. Centralized control in Software-Defined Networking (SDN) provides a framework for centralized network optimizations, such as optimal flow routing. However, adaptive flow routing that responds to dynamic traffic load variations requires sophisticated models. This study adopts a reinforcement learning-based approach, where direct flow routing generates a state-action pair space that grows with the number of active flows, leading to a potential scalability issue. This problem primarily arises from the large memory space required to store the Q-table containing flow-specific state-action pairs, as well as the prolonged learning time

needed to reach convergence, which involves exploring this vast state-action space. On the other hand, the computational overhead per step remains low, as it only requires updating Q-values at each step. To address this scalability challenge, we propose a distributed approach employing multiple SDN controllers, where each controller manages a smaller subset of network nodes and flows. Consequently, each controller operates with a smaller Q-table and a reduced state-action space. Each controller independently optimizes flow routing within its manageable subnetwork using reinforcement learning, with every Autonomous System (AS) improving routing within its own domain. Experimental results demonstrate that this approach achieves significant performance improvements: the average convergence time for multi-controller setups is only 10 steps, compared to 40.5 steps in the single centralized controller case. Furthermore, the average latency decreased from 31.30 ms (in the single-controller system) to 23.14 ms and 23.99 ms in the multi-controller systems. These results confirm that distributing control responsibilities effectively reduces the complexity of the learning space, accelerates convergence, and enhances flow routing efficiency, making the proposed solution suitable for large-scale and dynamic networks.

**Keywords**: Software-Defined Networking (SDN), Flow Routing, Reinforcement Learning, Scalability.

**1.مقدمة:**

يُعد توجيه التدفقات (Flow Routing) مشكلة أساسية في شبكات تبديل الحزم، حيث يجب أن تتبع الحزم الخاصة بتدفق معين – كتدفق من مضيف مصدر محدد إلى مضيف وجهة محدد أو تدفق بروتوكول التحكم في النقل – نفس مسار التوجيه (Routing Path) عبر شبكة عقد تبديل الحزم (Packet–Switching Nodes) ومع ذلك، يمكن أن تتبع مسارات مختلفة حتى بين نفس زوج المبدلات (المصدر–الوجهة) حيث تكون مبدلة المصدر هي مبدلة الدخل Ingress) (Switchلنطاق التوجيه (Routing Domain) وتكون مبدلة الوجهة هي مبدلة الخرج (Egress Switch)لنطاق التوجيه للتدفقات المعتبرة بهدف تحسين أداء الشبكة [1]. وبالتالي يختلف توجيه التدفقات (Flow Routing) بشكل جوهري عن توجيه المسارات الكلاسيكي (Classical Path Routing)القائم فقط على مضيف الوجهة كما في بروتوكول الإنترنت (Internet Protocol – IP) أو على زوج المبدلات المصدر–الوجهة. يمكّن توجيه التدفقات مجموعة واسعة من التكيفات لتحسين الأحمال على وصلات الشبكة (Network Links) أي لإجراء هندسة مرور فعالة بهدف تحسين مقاييس أداء الشبكة المختلفة [1] . ومع أن توجيه التدفقات يوفّر مرونة كبيرة في تحسين أداء الشبكة، إلا أن تحقيق هذه المرونة في بيئات ديناميكية وكبيرة الحجم يواجه تحديات جوهرية، خاصةً عند الاعتماد على تقنيات التعلّم الذكي مثل التعلّم المعزز (Reinforcement Learning). ففي الشبكات المعرفة برمجياً (SDN)، حيث يُدار التوجيه من خلال متحكم مركزي، يؤدي تزايد عدد التدفقات النشطة إلى نمو هائل في فضاء الحالات (State Space) وفضاء الأفعال (Action Space)، ما يُعقّد عملية التعلّم ويُبطئ التقارب، بل وقد يجعل النظام غير قابل للتوسع. ويزداد الوضع سوءًا عندما يُطلب من المتحكم اتخاذ قرارات توجيه في الوقت الفعلي لتدفقات متنوعة من حيث الحجم، الأولوية، ومتطلبات زمن الوصول. وبالتالي، يبرز التحدي البحثي المركزي: كيف يمكن تحقيق توجيه تدفقات ذكي وقابل للتوسع، يوازن بين كفاءة التعلّم، سرعة الاستجابة، وتحسين أداء الشبكة الشامل، دون التضحية بقابلية التوسع أو جودة الخدمة؟

## 2.هدف البحث

يهدف هذا البحث إلى معالجة مشكلة قابلية التوسع من خلال استخدام عدة متحكمات SDN بحيث يكون لكل متحكم مجال خاص به و بالتالي يعمل كل متحكم على اتخاذ قرار التوجيه لعدد أقل من العقد والتدفقات، الأمر الذي ينتج عنه فضاء حالة–إجراء أصغر لاستكشافه مما يسهم بالحصول على جدول Q أصغر. نهدف أيضاً إلى تحسين توجيه التدفق عن طريق استخدام التعلم المعزز الذي يعمل على اختيار المسار الأفضل كما سنرى لاحقاً.

## 3.مواد وطرق البحث

تم استخدام محاكي Mininet، وهو برنامج محاكاة يعتمد على Linux من أجل نمذجة الشبكات المعرفة برمجياً [2]. يعمل Mininet على تشغيل وإدارة مجموعة من الأجهزة المضيفة (Hosts) والمبدلات (Switches) والموجهات (Routers) والوصلات (Links) ووحدات التحكم (Controllers) باستخدام المحاكاة الافتراضية (Virtualization) لجعل نظام واحد يبدو وكأنه شبكة كاملة. يتضمن Mininet واجهة أوامر خاصة بالشبكة (Network–Aware CLI)، كما يوفر Mininet واجهة لبرمجة التطبيقات (Python API) لإنشاء الشبكة واختبارها [2]. يحتوي Mininet على واجهة مستخدم بسيطة يوفرها الملف النصي (Script) [4].

كما استخدمنا أداة iPerf وهي أداة مفتوحة المصدر ومجانية تُستخدم لقياس وتحليل أداء الشبكات، خاصة في قياس عرض النطاق الترددي (Bandwidth – BW) بين نقطتين على الشبكة، بالإضافة إلى قياس فقدان الحزم (Packet Loss) والتأخير (Delay) والتنوع في التأخير (Jitter). تعتمد iPerf على نموذج العميل والخادم (Client–Server Model)، حيث يُشغل وضع الخادم (Server Mode) على جهة، ويتصل به وضع العميل (Client Mode) ليبدأ اختبار نقل البيانات باستخدام بروتوكولات متعددة مثل بروتوكول التحكم في النقل (Transmission Control Protocol – TCP) وبروتوكول بيانات المستخدم ( User

Datagram Protocol – UDP) وبروتوكول التحكم في النقل المتقدم ( Stream Control Transmission Protocol – SCTP). تتيح الأداة ضبط عدة وسطاء مثل حجم الحزمة (Packet Size)، مدة الاختبار (Test Duration)، والنطاق الترددي المستخدم ( Used Bandwidth)، ما يمكن الباحثين من إجراء اختبارات دقيقة تحت ظروف مختلفة [5].

**4. الدراسات السابقة**

قام الباحثون في الدراسة [6] بتصميم خوارزمية توجيه في شبكات SDN تعتمد على التعلم المعزز باستخدام Q–learning مع مراعاة معايير جودة الخدمة (Quality of Service) QoS للوصلات الشبكية . تهدف الخوارزمية إلى الاستفادة الكاملة من وصلات الشبكة لتحديد المسار الأفضل. أظهرت النتائج التجريبية أنّ الخوارزمية قادرة على إيجاد المسار الذي يوفر جودة خدمة أفضل بدقة تقارب 100% بعد فترة تدريب معينة. من محدوديات خوارزمية Q–learningأن تصميم الميزات (features) المطلوبة يحتاج لتدخل يدوي، وهو أمر معقد وصعب التطبيق في بيئات الشبكات الحقيقية التي تتميز بتعقيد كبير في الخصائص. لمعالجة هذه المشكلة، اقترح الباحثون دمج الشبكات العصبونية مع التعلم المعزز، بحيث يتم استبدال جدول Q التقليدي بدالة تقريبية يتم تدريبها، مما أدى إلى خوارزمية توجيه قائمة على التعلم المعزز العميق (Deep Q–learning) أظهرت نتائج التجارب أن خوارزمية التوجيه المعتمدة على Deep Q–learning تقدم أداءاً جيداً مع تحسينات ملموسة على جودة الخدمة واستغلال موارد الشبكة. هذا يشمل قدرة أفضل على التكيف مع بيئة الشبكة الديناميكية والمعقدة مقارنة بالخوارزميات التقليدية مثل Dijkstra و RIP.

بينت الدراسة [7] أن التعامل مع أحمال متنوعة ومتفاوتة يستلزم استخدام نموذج معقد، وبالتالي تم التركيز على تحقيق نظام تعلم معزز دون الاعتماد الشبكات العصبونية ( model–free RL scheme). تم تصميم نموذج QR–SDN، يقوم بإنشاء مسارات متعددة بين المصدر والوجهة،

مما يحقق تأخيرات تدفق أقل بكثير. أشارت الدراسة إلى الجهود البحثية الإضافية لابتكار نهج قابل للتوسع مع زيادة حجم الشبكة.

قدمت الدراسة [8] نهجاً للتوجيه بعنوان التعلم المعزز والشبكات المعرفة برمجياً للتوجيه الذكي ( Reinforcement Learning and Software-Defined Networking for Intelligent Routing - RSIR)، والذي يعتمد على الحاجة إلى إضافة مستوى المعرفة ( Knowledge Plane إلى الشبكة، والتي يتم تغذيتها بالبيانات التي تجمعها بواسطة مستوى الإدارة (Management Plane). وبشكل خاص، طوروا خوارزمية توجيه استباقية (proactive) مبنية على التعلم المعزز تعتمد على مقاييس حالة الرابط (link-state metrics) وتم تنفيذها في نموذج أولي مع مصفوفات حركة حقيقية. تمت مقارنة RSIR مع خوارزمية ديكسترا الكلاسيكية (classic Dijkstra's algorithm)، والتي تستند إليها معظم بروتوكولات التوجيه. أظهرت النتائج أن RSIR يحصل على المزيد من المسارات الأقصر ويستطيع توازن الحمل بشكل أفضل، وبالتالي تقليل التأخيرات الإجمالية. من توصيات هذه الدراسة الحاجة إلى تطوير النهج إلى التعلم المعزز العميق.

قدمت الدراسة [9] تقنية Deep Q-Routing (DQR)، والتي تستخدم شبكة Q العميقة (dueling deep Q-network) مع إعادة تشغيل التجربة ذات الأولوية ( prioritised experience replay) لحساب مسار أي طلب زوج مصدر-وجهة مع وجود مقاييس QoS متعددة، مثل التأخير (delay)، عرض النطاق الترددي (bandwidth) أو الفقدان (loss). تمت المقارنة مع طرق التعلم الحالية الأخرى للتوجيه المباشر على الإنترنت ( greedy online routing)، وأظهرت نتائج أفضل من حيث الفقدان (loss) وتكلفة المسار (path cost)، مع الحفاظ على أفضل عرض نطاق ترددي معظم الأوقات وتأخير معقول.

اقترحت الدراسة [10] نهجاً مبنياً على الشبكة العصبونية العميقة DNN في نشر شبكات SDN/OSPF الهجينة. تقوم وحدة تحكم SDN بأداء توجيه موفر للطاقة وأداء محسّن مع

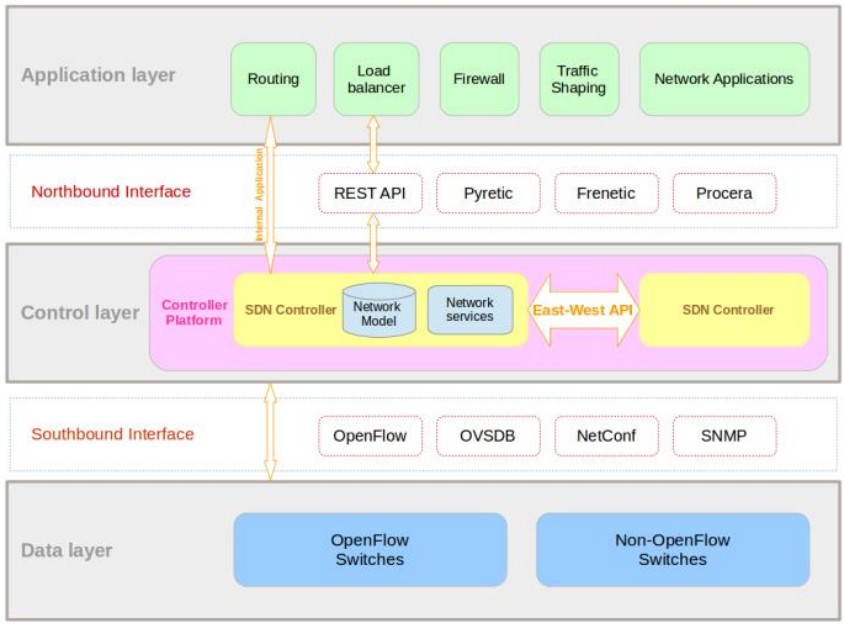
ضمانات جودة الخدمة QoS. تضمّن النموذج المعتمد عليه وحدة تعلم آلي ووحدة التعلم المعزز العميق DRL. يتكوّن التعلم الآلي من شبكة LSTM تقوم بالتنبؤ بتدفق الحركة باستخدام مجموعات بيانات السلاسل الزمنية، والتي تستخرج تقلبات وفترات تكرار بيانات الشبكة قصيرة المدى لضمان تنبؤ تدفق الحركة والتوجيه الموفر للطاقة مع أداء QoS مضمون. تقوم وحدة DRL بالتعلم من البيانات التاريخية الموجودة والتعلم التكراري من التفاعل مع إعدادات الشبكة المحددة.

بناءاً على ما سبق ذكره من دراسات مرجعية وبالإشارة إلى توصيات هذه الأبحاث، تم تحديد مشكلة البحث المتمثلة في تحسين التدفق في شبكات SDN بشكل يراعي قابلية التوسع والكفاءة في اختيار المسارات عن طريق تقسيم الشبكة إلى عدة مجالات واستخدام التعلم المعزز.

### 5.بنية شبكة المعرفة برمجياً Software-Defined Networking

تقترح مبادرة الشبكات المُعرَّفة برمجيا (SDN)، التي تقودها مؤسسة الشبكات المفتوحة (ONF) (the Open Networking Foundation)، بنية مفتوحة جديدة لمواجهة تحديات الشبكات الحالية، مع إمكانية تسهيل أتمتة عمليات تهيئة الشبكة، والأفضل من ذلك، برمجة الشبكة بالكامل. وعلى عكس البنية التقليدية الموزعة للشبكات، حيث تكون أجهزة الشبكة مغلقة ومتكاملة رأسياً (vertically-integrated)، وتدمج البرمجيات مع الأجهزة، فإن بنية SDN (كما في الشكل (1)) ترفع مستوى التجريد (abstraction) عن طريق فصل مستوى البيانات عن مستوى التحكم في الشبكة. وبهذه الطريقة، تصبح أجهزة الشبكة مجرد مبدلات بسيطة لإعادة التوجيه، بينما تتركز كل منطقية التحكم (control logic) في وحدات تحكم برمجية، مما يوفر إطاراً برمجياً مرناً لتطوير التطبيقات المتخصصة ونشر الخدمات الجديدة [12].

يُعتقد أن هذه الجوانب في بنية SDN تبسط وتحسن إدارة الشبكة من خلال إتاحة إمكانية الابتكار، وتخصيص السلوكيات، والتحكم في الشبكة وفقًا لسياسات عالية المستوى يتم التعبير عنها كبرامج مركزية، مما يسمح بتجاوز تعقيدات تفاصيل الشبكة على المستوى المنخفض والتغلب على المشاكل البنيوية الأساسية .يُضاف إلى هذه الميزات قدرة SDN على التعامل بسهولة مع عدم تجانس البنية التحتية وذلك بفضل التجريد الذي توفره الواجهة البينية الجنوبية (Southbound interface) في SDN [12].



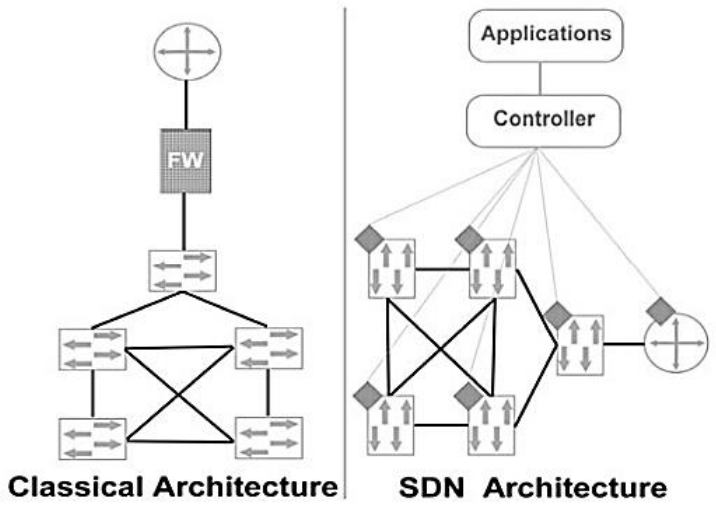

**الشكل (1): بنية الشبكات المُعرَّفة برمجيا الموزعة ذات الثلاث طبقات [12].**

### 6.مقارنة بين الشبكات المعرفة برمجياً والشبكات التقليدية

في البنية التحتية التقليدية (كما في الشكل (2))، يتطلب تنفيذ الشبكة وتهيئتها واستكشاف أخطائها وإصلاحها تدخل مهندسي شبكات وأنظمة ذوي مهارات تقنية

عاليـــة، بالإضـــافة إلـــى التكـــاليف التشـــغيلية المرتفعـــة المرتبطـــة بتـــوفير وإدارة الشـــبكات الكبيـرة متعـددة المـوردين. وفـي الواقـع، إن تنـوع عناصـر الشـبكة وتعقيـدها [12] يجعـلان صـيانتها باهظـة التكلفـة، والبنيـة التحتيـة الأساسـية أقـل موثوقيـة فـي حالـة حـدوث أعطـال متكررة للشبكة، خاصة إذا لم يتم تجهيز خطط احتياطية ضمن البنية التحتية.

نظـراً لأن SDN تفصـل قـرارات التوجيـه وإعـادة التوجيـه الخاصـة بعناصـر الشـبكة (مثـل الموجّهـات والمبـدلات ونقـاط الوصـول) عـن مسـتوى البيانـات (data plane)، فـإن إدارة الشـبكة تصـبح أقـل تعقيـداً (كمـا موضـح فـي الجـدول (1)) الـذي يوضـح تفـوق شـبكات SDN علـى الشـبكات التقليديـة؛ وذلـك لأن مسـتوى الـتحكم (control plane) يتعامـل فقـط مـع المعلومـات المتعلقـة بهيكليـة الشـبكة المنطقيـة (logical network topology)، وتوجيـه حركـة المـرور، ومـا إلـى ذلـك. فـي المقابـل، يقـوم مسـتوى البيانـات بتنسـيق حركـة مـرور الشـبكة وفقًـا للتهيئـة التـي تـم إعـدادها فـي مسـتوى الـتحكم فـي بنيـة SDN، تكـون عمليـات الـتحكم مركزيـة فـي وحـدة تحكـم (Controller) هـي التـي تفـرض سياسات الشبكة [13].




**الشكل (2): بنية الشبكات المعرفة برمجيا مقابل بنية الشبكات التقليدية [13].**

**جدول (1): مقارنة بين الشبكات المُعرَّفة بالبرمجيات (SDN) والشبكات التقليدية [13].**

| الخصائص | البنية التقليدية | بنية SDN |
|---|---|---|
| القابلية للبرمجة | | ✓ |
| التحكم المركزي | | ✓ |
| التهيئة والعُرضة للخطأ | ✓ | |
| التحكم المعقد بالشبكة | ✓ | |
| مرونة الشبكة | | ✓ |

| | | |
|---|---|---|
| أداء مُحسَّن | | ✓ |
| سهولة التنفيذ | | ✓ |
| تهيئة فعّالة | | ✓ |
| إدارة مُحسَّنة | | ✓ |

### 7.مشاكل توجيه التدفق في SDN

تتمثل التحديات والقضايا ذات الأهمية القصوى في توجيه حركة المرور ضمن بيئة SDN فيما يلي [14] :

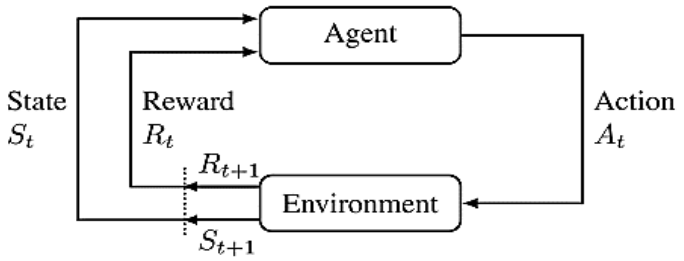
- **قابلية التوسع** (Scalability) :إن قدرة شبكات الـ SDN على التعامل مع أعداد هائلة من الأجهزة وتدفقات حركة المرور له أمر بالغ الأهمية. وهذا يطرح تحدياً، حيث يجب على وحدات تحكم SDN إدارة كميات كبيرة من البيانات وتحديد المسارات بسرعة.
- **التكيف في الوقت الفعلي مع تغيرات الطوبولوجيا** Real-time adaptation to topology) (shifts : يجب على شبكات SDN الاستجابة الفورية للتغيرات الديناميكية في بنية الشبكة. وهذا أمر حتمي، حيث يجب على وحدات تحكم SDN إعادة توجيه حركة المرور بسرعة لتجاوز الاتصالات الفاشلة أو العُقد المزدحمة.
- **تلبية متطلبات جودة الخدمة (QoS) المتنوعة** : يعد الالتزام بمتطلبات جودة الخدمة المحددة لكل تطبيق أمراً ضرورياً لشبكات .SDN. يجب على وحدات تحكم SDN تحديد أولويات حركة المرور وتوجيهها بفعالية، بما يتماشى مع الاحتياجات الخاصة بكل تطبيق.
- **اعتبارات الأمن والخصوصية المرتفعة** :إن ضمان أمن وسرية بيانات المستخدم داخل شبكات SDN يعتبر أمر بالغ الأهمية. إن التحكم المركزي والبيانات المركزية في شبكات SDN يجعلانها عرضة للانتهاكات المحتملة.

- **الاستخدام الأمثل للموارد** :يعد الاستغلال الفعال لموارد الشبكة مطلباً أساسياً لشبكات SDN.يجب على وحدات تحكم SDN صياغة قرارات توجيه تقلل من الازدحام وتعظم من استخدام عرض النطاق الترددي.

على الرغم من هذه التحديات، تُعد SDN تقنية واعدة لديها القدرة على إحداث ثورة في طريقة إدارة الشبكات. ومع نضوج هذه التقنية، من المرجح أن يتم التعامل مع هذه التحديات، وستصبح SDN حلاً أكثر جدوى لمجموعة واسعة من الشبكات.

## 8. تحسين التوجيه القائم على التعلم المعزز:

تُطبَّق خوارزميات التعلم المعزز (RL) بشكل عام لحل مسائل اتخاذ القرار. وعند تطبيقها لتحسين التوجيه، تؤدي وحدة التحكم دور **الوكيل (agent)**، بينما تمثل الشبكة **البيئة (environment)** ويتألف **فضاء الحالة (state space)** من حالات الشبكة وحركة المرور. أما **الإجراء (action)** فهو حل التوجيه المختار. وتُعرَّف **المكافأة (reward)** بناءً على مقاييس التحسين، مثل تأخير الشبكة كما هو موضح في الشكل (3) [15].




**الشكل (3): آلية عمل التعلم المعزز [15].**

عند كل خطوة زمنية t، يراقب الوكيل حالة $S_t$، ويختار إجراءً $A_t$ من فضاء الإجراءات A، ويستقبل مكافأة فورية $R_t$ (تشير إلى مدى جودة أو سوء هذا الإجراء)، ثم ينتقل إلى الحالة التالية $S_t$ +1. يتمثل هدف الوكيل في تعلم **سياسة السلوك المثلى** $\pi$ **(optimal behavior policy)** ، وهي عبارة عن ربط مباشر من فضاء الحالة S إلى فضاء الإجراءات A أي ($\pi$: S → A)، وذلك بهدف تعظيم المكافأة المتوقعة طويلة الأمد [16] .

انطلاقًا من سياسة السلوك $\pi$، يستطيع الوكيل تحديد أفضل إجراء مقابل لحالة معينة. في التعلم المعزز، تُستخدم **دالة القيمة (value function)** لحساب المكافأة طويلة الأمد لإجراء معين في حالة معينة. إن دالة القيمة الأشهر هي **دالة-Q (Q-function)**، والتي تستخدمها خوارزمية **تعلم Q (Q-learning)** لتعلم جدول يخزن جميع أزواج الحالة-الإجراء (state-action pairs) ومكافآتها طويلة الأمد[16].

### 9. خوارزمية Q-LEARNING

تُعد (Q-learning) خوارزمية تعلم المعزز غير معتمدة على النموذج (model-free)، تهدف إلى تعليم الوكيل سياسة الإجراءات التي يجب اتباعها وفقاً للحالة والملاحظات المستقاة من البيئة. وبصفته أسلوباً غير معتمد على النموذج، فإنه لا يستخدم احتمالية الانتقال (transition probability) [17] .

يعمل هذا الأسلوب ضمن إطار عمليات ماركوف لاتخاذ القرار *Markov Decision Process* (MDP)، حيث يسعى لإيجاد سياسة مثلى (optimal policy) من خلال تعظيم القيمة المتوقعة للمكافأة التراكمية (cumulative reward) المحسوبة على مدار جميع الخطوات المتتالية، بدءاً من الحالة الحالية. وفي الوقت الحاضر، يُشكِّل هذا الأسلوب خط أساس (baseline) تُقارَن به أساليب التعلم المعزز الأخرى كما هو موضح في المعادلة (1) [17].

$$Q(S_t, A_t) \leftarrow (1-\alpha)Q(S_t, A_t) + \alpha\left(R_{t+1} + \gamma \max_{a\in A} Q(S_{t+1}, a)\right) \quad [17] \quad (1)$$

حيث يحدد **معدل التعلم** (**learning rate**) $\alpha$ مدى سرعة استيعاب قيم Q المكتسبة حديثًا. ويمثل **معامل الخصم** (**discount rate**) $\gamma$ مدى أهمية المكافآت المستقبلية المتوقعة. يتم تمثيل المكافآت المستقبلية المتوقعة بالمعادلة $\max_{a\in A} Q(S_{t+1}, a)$، والتي تمثل بشكل أساسي أقصى مكافأة يمكن الحصول عليها عند اتخاذ الإجراء a ذي القيمة الأعلى.

### 1.9 تصميم فضاء الحالة (STATE-SPACE DESIGN)

بفرض لدينا الشبكة G(V, E) حيث E هي مجموعة من الحواف التي تربط مجموعة الرؤوس V. نركز على تدفقات الاتصال الأحادية البث ( unicast communication flows)، أي التدفقات التي تنقل البيانات من مرسل معين إلى مستلم واحد. يشار إلى نقل البيانات إلى طبقة التطبيق أو النقل من مرسل معين (مضيف المصدر) $s_f$ إلى مستلم معين (مضيف الوجهة) $d_f$ بالتدفق f. نرمز إلى مجموعة التدفقات بالرمز F. نفترض أن التدفق f ينقل معدل حركة معين $R_f$ من مضيف المصدر sf إلى الشبكة. المسار $(Ps_f,d_f)$ هو تسلسل من الرؤوس P = (v1, … , vn) من مجموعة جميع المسارات الممكنة $Ps_f,d_f = \{ Ps_f,d_{f,1}, Ps_f,d_{f,2}, \ldots\}$ P $\in$ التي تربط مضيف المصدر $s_f$ بمضيف الوجهة $d_f$، حيث يمكن تحديد مجموعة $(Ps_f,d_f)$ بواسطة خوارزمية بحث في الرسم البياني، مثل البحث بالعمق (Depth-First Search – DFS) [18].

يراقب وكيل التعلم المعزز، الذي قد يعمل في وحدة تحكم SDN، البيئة (أي الشبكة) من خلال قياس المؤشرات الرئيسية للأداء المطلوبة، مثل زمن الاستجابة أو عرض النطاق الترددي، في خطوات زمنية منفصلة.... t = 0, 1,2, الملاحظة تتكون من حالة البيئة $S_t$ من مجموعة الحالات

S ={$S^1$, $S^1$, …} ومكافأة $R_t \in R \subset R$. نحدد الحالة $S_t$ لتتكون من جدول، يحتوي على المسار المحدد حالياً P لكل تدفق f:

$$S_t = \begin{cases} f_{s_1,d_1} : P_{s_1,d_1,t} \\ \vdots \\ f_{s_i,d_i} : P_{s_i,d_i,t} \end{cases} \quad [7] \quad (2)$$

يتكون فضاء الحالة من قاموس من المبدلات والقيم مع التدفقات كمفاتيح والمسار الحالي كقيمة لكل مفتاح. نلاحظ أن هذا القاموس هو مجرد تطبيق ممكن واحد لفضاء الحالة. يمكن أيضاً تمثيل الحالات كقائمة، يمكن ربطها مباشرة بمدخلات الشبكة العصبية المستخدمة في التعلم Q العميق. لقد فضلنا القاموس مع التدفقات كمفاتيح والمسارات كقيم، حيث أن هذا تمثيل مباشر لفضاء الحالة من وجهة نظر تنفيذ البرمجة [7].

### 2.9 تصميم فضاء الإجراء (ACTION–SPACE DESIGN)

اعتماداً على الحالة $S_t$ ومكافأتها المقابلة $R_t$ ، يتم اختيار إجراء A $A_t \in$ (حيث قد تعتمد مجموعة الإجراءات الممكنة A بشكل عام على الحالة $S_t$). مجموعة الإجراءات A = { $A_{t,1}$, $A_{t,2}$, …} تُحدد بواسطة مجموعة المسارات الممكنة، بما في ذلك المسار الحالي، أي A = $Ps_f,d_f$ للتدفق f. يتم اختيار أحد هذه المسارات الممكنة إما لاستبدال المسار الحالي أو الاحتفاظ به. وبالتالي، يغير الإجراء القيمة، أي المسار الحالي، لمبدل، أي تدفق.

ف إجراء $A_t$ المطبق على تدفق واحد بواسطة: [7] (3)

$$A_t = \{f_{s_1,d_1} : P_{s_1,d_1,t} \Rightarrow P_{s_1,d_1,t+1}\}$$

الآن يبقى السؤال عما إذا كان يجب تغيير تدفق واحد أو عدة تدفقات بإجراء واحد. يمكننا تغيير تدفق واحد في خطوة زمنية، أي إجراء تغيير تدفق واحد كما هو محدد في

المعادلة (4). بدلاً من ذلك، يمكننا تغيير جميع التدفقات في خطوة زمنية، أي اتخاذ الإجراء:

$$A_t = \begin{cases} \{f_{s_1,d_1}: P_{s_1,d_1,t} \Rightarrow P_{s_1,d_1,t+1}\}, \\ . \\ . \\ . \\ \{f_{s_i,d_i}: P_{s_i,d_i,t} \Rightarrow P_{s_i,d_i,t+1}\}, \end{cases} \quad [7] \ (4)$$

والتي نشير إليها بتغيير مباشر (Direct Change). من الواضح، كما تشير المعادلتان (3) و(4)، أن تصميم الإجراء $A_t$ له تأثير على حجم فضاء الإجراء $|A|$. عيب نهج التغيير المباشر هو أن فضاء الإجراء يتناسب مع حاصل ضرب أعداد المسارات الممكنة للتدفقات؛ بالمقابل، يتناسب فضاء إجراء تغيير تدفق واحد مع مجموع أعداد المسارات الممكنة للتدفقات. من ناحية أخرى، يسمح التغيير المباشر بالتبديل المباشر في خطوة زمنية واحدة إلى الحالة المرغوبة (تكوين التوجيه) من أجل تحقيق أداء توجيه تدفق أعلى [7].

### 3.9 تصميم المكافأة (REWARD DESIGN)

تستخدم المكافأة $R_t$ +1 لقياس مدى جودة إجراء $A_t$ في حل مشكلة توجيه التدفقات. مدعومين بالاهتمام المتزايد في شبكات زمن الاستجابة المنخفض [19]، [20]، تأخذ تقييماتنا في هذه الدراسة في الاعتبار زمن الاستجابة للمكافأة. أيضاً، فإن الازدحام يزيد بشكل عام زمن الاستجابة، ولكن ليس معدل البت الناقل المستهلك. يتطلب اعتبار معدل النقل للمكافأة معرفة بمعدل البت الناقل المطلوب لكل تطبيق. بدون معرفة بمتطلبات الأجهزة المرسلة، سيكون من غير الواضح ما إذا كان تغيير معدل النقل يعود إلى قرار توجيه سيئ أو فقط لأن الجهاز المرسل قد خفض معدل البت الناقل. إذا كان يجب اعتبار مقاييس أداء متعددة، فيمكن استخدام صيغة موزونة كما هو مقترح في [21].

تتكون مكافأتنا المقترحة $R_t$ من مجموع أزمن الاستجابة $L_f$ على طول المسارات الحالية $Ps_{f,d_f}$ للتدفقات $f \in F$. من أجل إعطاء وزن ثقيل نسبياً للقيم المتطرفة نستخدم الجذر التربيعي للمتوسط:

$$R_t = -\sqrt{\left(\frac{\sum\_\forall f \epsilon F \; L_f^2}{|F|}\right)} \qquad [7]\ (5)$$

نلاحظ أن الإشارة السالبة مطلوبة لأن وكيل التعلم المعزز يسعى لتعظيم مكافأته؛ ومع ذلك، فإن زمن الاستجابة الأعلى أقل رغوبة.

### 4.9 استراتيجية الاستكشاف (EXPLORATION STRATEGY)

في التعلّم المعزز، يواجه الوكيل (Agent) تحدي تحقيق توازن بين الاستكشاف (Exploration) — أي تجربة إجراءات جديدة لاكتساب معرفة عن البيئة — والاستغلال (Exploitation) — أي استخدام المعرفة الحالية لتعظيم المكافأة. ولتحقيق هذا التوازن، تُطبَّق استراتيجيات استكشاف معيارية. حيث تم تطبيق استراتيجية Softmax بمعامل درجة الحرارة τ، بحيث أن درجة حرارة τ منخفضة تفضل الاستغلال (أي، يتم اختيار الإجراء ذو أعلى قيمة Q أكثر غالباً)، ودرجة حرارة τ عالية تؤدي إلى طابع استكشافي لاقتناء معرفة جديدة.

بما أن قيم Q لدينا تُهيّأ بـ −∞، فيتم استخدام المعادلة التالية إلى:

$$Pr\{A_t = a\} = \frac{exp\left[-\frac{1}{Q(s,a)\cdot\tau}\right]}{\sum_{(b\in A)} exp\left[-\frac{1}{Q(s,b)\cdot\tau}\right]} \qquad [7]\ (6)$$

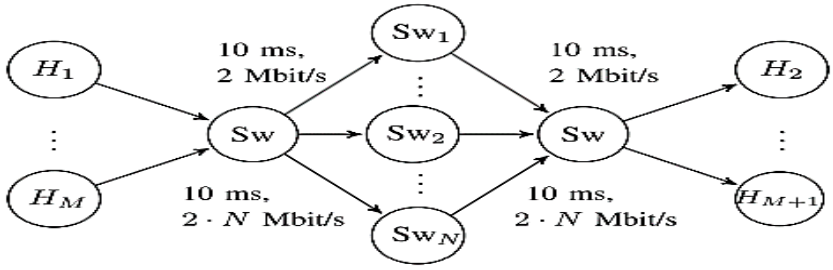
من أجل محاكاة سلوك Softmax لنطاقات القيم السالبة. لا تصل قيم Q أبداً إلى الصفر، في المعادلة (6)، الاحتمالات غير صفرية للتهيئة $Q(s,a) = -\infty$. سيؤدي

هذا إلى تجربة جميع تركيبات الحالة–الإجراء، لأن الاحتمالات لـ Q(s,a) = 0 هي الأعلى. هذا الاستكشاف القسري لجميع تركيبات الحالة–الإجراء، ومع ذلك، سيتناقض مع الفكرة أن الاستكشاف يُتحكم به بواسطة درجة الحرارة τ [7].

## 10. القابلية للتوسع (SCALABILITY)

القابلية للتوسع تمثل تحدياً رئيسياً في التوجيه المستند إلى التعلم المعزز. ندرس القابلية للتوسع مع الطوبولوجيا في الشكل 4 ذات N مفتاح وسيط وM تدفق، بحيث تدعم روابط المبدلة الوسيطة Swi سرعة نقل تبلغ 2 · i ميغابت/ثانية ولكل تدفق $f_i$ من المضيف $H_i$ إلى $H_{i+1}$ طلب عرض نطاق ترددي يبلغ 2 · i ميغابت/ثانية (ناقص هامش 250 كيلوبت/ثانية). يمكن استنتاج التوزيع الأمثل للتدفقات بسهولة لهذه الطوبولوجيا، أي $f_1$ عبر Sw1، $f_i$ عبر Swi، وهكذا. بالنسبة لهذه الطوبولوجيا، يمكننا حساب حجم جدول Q لتغيير تدفق واحد كما يلي [7]:

$$\underbrace{|Q(s,a)|}_{\text{(Size of Q-table)}} = \underbrace{N^{\wedge}M}_{\text{(Number of states)}} * \underbrace{([(N-1)\cdot M+1])}_{\text{(Number of actions)}} \quad [7] \ (7)$$




**شكل 4: طوبولوجيا تقييم قابلية التوسع مع N من المبدلات الوسيطة و M من التدفقات**

### 11. متحكم Ryu في شبكاتSDN

متحكم Ryu هو إطار عمل مفتوح المصدر مخصص للتحكم البرمجي في الشبكات (SDN)، ويسمح للمطورين بكتابة تطبيقات شبكية متقدمة تسهّل إدارة الشبكات ديناميكيًا ومرونة أكبر من الشبكات التقليدية [22] [23] [24].

- **تعريف** Ryu :هو متحكم شبكي موجه لبيئة SDN ، مُبرمج بالكامل بلغة Python، ويوفر مجموعة من الواجهات البرمجية Application Programming Interface (APIs) تسهّل للمطورين بناء حلول لإدارة وتوجيه البيانات في الشبكة عبر بروتوكولات مثل OpenFlow وغيرها.
- **الوظيفة الأساسية** : إصدار قرارات ديناميكية حول كيفية مرور البيانات بين أجهزة الشبكة (المفاتيح – السويتشات)، عبر إرسال تعليمات مباشرة لها.
- **الدعم البرمجي** : يدعم Ryu كتابة تطبيقات شبكية بسهولة بسبب بساطة Pythonومرونة البرمجة الحدثية(event-driven) ، ويمكن تشغيل التطبيق عبر أداةryu-manager [23][25] .

المميزات:

- **واجهة برمجة تطبيقات واضحة** :يُبسط عمليات تطوير التطبيقات الشبكية، ويحتوي على دوال ووحدات جاهزة للتعامل مع أحداث الشبكة ورسائل OpenFlow [23][25].
- **تعدد البروتوكولات** :يدعم نسخ متعددة من OpenFlow (1.0 حتى 1.5)، ويدعم بروتوكولات أخرى مثل Netconf و OF-config [23].
- **مرونة الاختبار والبحث** :مثالي للبيئات البحثية والأكاديمية بفضل سهولة إنشاء ومحاكاة الشبكات باستخدام أدوات مثل Mininet.

1. كل تطبيق Ryu هو سكربت Python يرث من الصنف RyuApp.
2. يمكن لأي تطبيق مراقبة الأحداث الشبكية (مثل وصول رزمة، تغيير حالة منفذ).
3. يُرسل Ryu الرسائل والتعليمات للمفاتيح مباشرة باستخدام OpenFlow [25][23].

## 12.مقاييس الأداء (PERFORMANCE METRICS)

نستخدم المقاييس التالية في تقييمنا، وهي [7]:

### 1.12 الوقت حتى التقارب (Time till convergence)

يصف الوقت حتى التقارب مدى سرعة عثور وكيل التعلم المعزز على حالة زمن استجابة منخفض. يجب ألا يستغرق هذا البحث عن حالة زمن استجابة منخفض وقتاً طويلاً، لأنه في الواقع وفي محاكاتنا، لا يمكن إجراء تسريع (على سبيل المثال، من خلال أجهزة حوسبة أسرع)؛ بل إن مدة البحث مرتبطة بشكل جوهري بتصميم خوارزمية البحث، أي خوارزمية التعلم المعزز ومدة خطوة زمنية (أي الفترة الزمنية لتكرار تعلم واحد). لذلك، يجب على وكيل التعلم المعزز أن يتعلم بسرعة وكفاءة قدر الإمكان. لا يوجد تعريف شائع للتقارب في التعلم المعزز. من أجل تقييم أوقات التقارب كمياً، قمنا أولاً بتنعيم أزمن الاستجابة المتوسطة المقيسة للتدفقات بمتوسط متحرك لـ N =10 خطوة متتالية. بعد ذلك، حسبنا الفروق المحدودة لأزمنة الاستجابة المتوسطة المُنعّمة. إذا كانت التقلبات، أي الفروق المحدودة لأزمن الاستجابة المتوسطة المُنعّمة، أصغر من قيمة عتبة محددة تبلغ 0.4 مللي ثانية/خطوة، فإننا نعتبر النظام متقارباً [7].

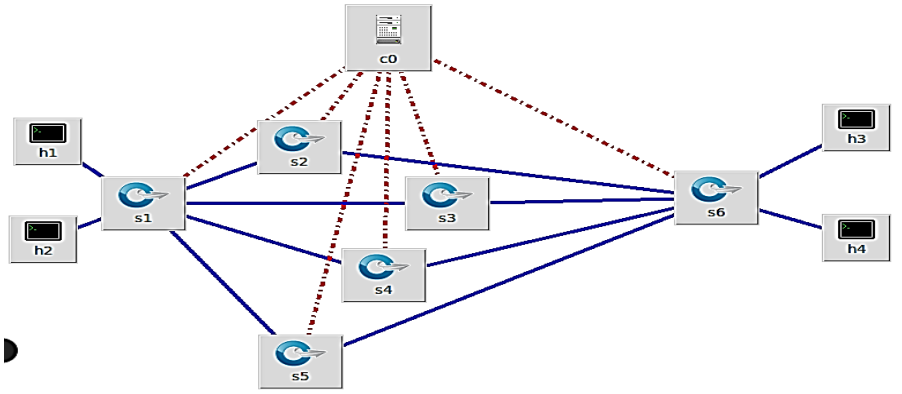
### 2.12 متوسط زمن استجابة التدفق (AVERAGE FLOW LATENCY)

يُعد زمن الاستجابة مؤشراً مفيداً لأداء الشبكة حيث يكشف عن الازدحام، الذي لا يزيد زمن الاستجابة فحسب، بل يقلل من معدل النقل أيضاً. وبما أنه في الشبكة الحقيقية لن نعرف أزمنة الاستجابة التي تمر بها الحركة الفعلية، فإننا نقيس أزمنة استجابة التدفقات عبر حزم الاستطلاع. نحدد متوسط زمن الاستجابة كمتوسط زمن استجابة التدفق غير الموزون عبر جميع التدفقات على طول مساراتها respective من المصدر إلى الوجهة [7].

### 13.التنفيذ العملي والنتائج

سوف نقوم بإجراء اختبار وذلك لبنية شبكة مشابهة للشكل 4 كما هي ظاهرة في الشكل (5) وذلك باستخدام متحكم واحد ثم سنقوم بفصل الشبكة الى نطاقين وذلك باستخدام متحكمين بحيث كل متحكم يكون مسؤول عن نطاق.

### 1.13 السيناريو الأول: شبكة باستخدام متحكم واحد




**الشكل 5: شبكة باستخدام متحكم واحد مسؤول عن إدارة التدفقات لكامل الشبكة.**

تم استخدام الشيفرة التالية لتوليد تدفق باستخدام أداة iperf بين كل زوج متقابلين من الأجهزة حيث ولد تدفق بين h1 ,h3 وتدفق بين h2,h4:

```python
def startIperf(host1, host2, amount, port, timeTotal, loadLevel):
    info(f"Killing old iperf clients on {host1.name}...\n")
    host1.cmd('pkill -f "iperf -c"') # pkill -f matches full command line
        # Give it a moment to terminate
    time.sleep(0.1)
    bw = float(amount) * (float(loadLevel) / float(10))
    print("Host {} to Host {} Bw: {}".format(host1.name, host2.name, bw))
    command = "iperf -c {} -u -p {} -t {} -b {}M &".format(host2.IP(), port, timeTotal, bw)

    host1.cmd(command)
```

host1: المضيف الذي سيُستخدم كـ client (يرسل البيانات).

host2: المضيف الذي سيُستخدم كـ server (يستقبل البيانات).

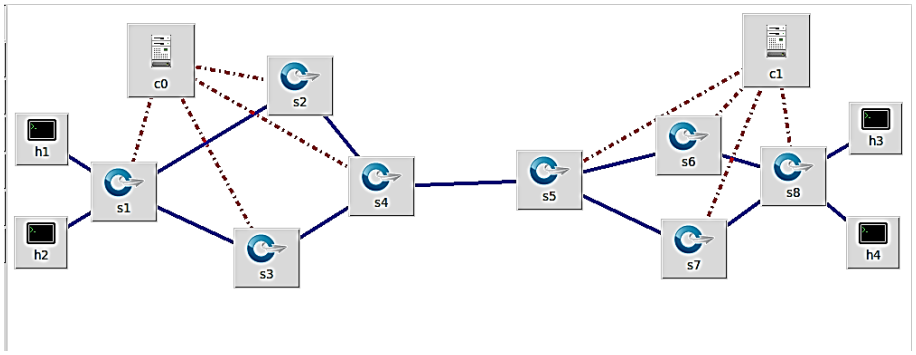
amount: قيمة أساسية لعرض النطاق الترددي (حسب المعادلة 2.0*j+1.75)، تُستخدم لحساب السرعة الفعلية.

port: المنفذ (port) الذي سيُستخدم في الاتصال (5001).

timeTotal: المدة الزمنية (بالثواني) وتم اختيار (20 دقيقة) التي سيُرسل فيها العميل البيانات.

loadLevel: مستوى الحمل (10)، يُستخدم لتعديل السرعة.

### 2.13 السيناريو الثاني: شبكة باستخدام متحكمين




**الشكل 6: شبكة باستخدام متحكمين كل متحكم مسؤول عن توجيه التدفق داخل نطاقه.**

وذلك باستخدام نفس اعدادات التدفق باستخدام متحكم واحد حيث يتم ارسال التدفق من النطاق الأول الى الثاني وبالعكس.

### 3.13 تقييم النتائج

1. الشبكة الأصلية (قبل التقسيم الشكل 5):

- الوصف:

في الشبكة الأصلية، جميع العقد (`s1`, `s2`, `s3`, `s4`, `s5`, `s6`) تُدار بواسطة متحكم وحيد (`c0`).

- المشكلة الرئيسية:

- مساحة الحالة-الفعل (`State-Action Space`):

مع زيادة عدد التدفقات (`flows`) النشطة، يزداد حجم المساحة الحالة-الفعل بشكل كبير.

- وقت التقارب (`Convergence Time`):

يتطلب استكشاف هذه المساحة الكبيرة وقتاً طويلاً للتعلم.

2. الشبكة بعد التقسيم إلى دومينات وذلك لتحقيق هدف البحث (الشكل 6) :

-الوصف:

تم تقسيم الشبكة إلى دومينين:

- الدومين الأول يحتوي على العقد (`s1`, `s2`, `s3`, `s4`) ويُدار بواسطة المتحكم (`c0`).

– الدومين الثاني يحتوي على العقد (`s8` ,`s7` ,`s6` ,s5) ويُدار بواسطة متحكم آخر (`c1`).

– توزيع المسؤوليات:

كل متحكم يركز على إدارة عدد أقل من العقد والتدفقات داخل نطاقه (`domain`)، مما يؤدي إلى:

– مساحة حالة–فعل أصغر:

بسبب عدد العقد والتدفقات الأقل، يتم تخفيض حجم مساحة الحالة–الفعل لكل متحكم.

– وقت تقارب أسرع:

مع مساحة حالة–فعل أصغر، يتم تحقيق الاستقرار (`convergence`) بسرعة أكبر.

3.السبب الأساسي لتقسيم الشبكة إلى دومينات هو تقليل مساحة الحالة–الفعل (`State–Action Space`):

– في الشبكة الأصلية:

– المساحة الحالة–الفعل كبيرة جدًا بسبب عدد العقد الكبير (s1, s2, s3, s4, s5, s6) وعدد التدفقات النشطة في مثالنا هما تدفقين.

– هذا يؤدي إلى مشكلة في التخزين (لتخزين الجدول Q) والتعلم (لاستكشاف المساحة الكبيرة).

– في الشبكة المقسمة:

– تم تقسيم الشبكة إلى دومينين، مما خفض عدد العقد التي يديرها كل متحكم.

– كل متحكم يتعامل مع مساحة حالة–فعل أصغر، مما يقلل من متطلبات التخزين ويزيد من سرعة التقارب.

وقت التقارب (`Convergence Time`):

– في الشبكة الأصلية:

– يحتاج المتحكم (`c0`) إلى استكشاف مساحة حالة–فعل كبيرة جدًا، مما يجعل وقت التقارب (`Time till Convergence`) طويلاً.

– في الشبكة المقسمة:

– كل متحكم يتعامل مع مساحة حالة–فعل أصغر، مما يؤدي لاستكشاف المساحة وتحقيق الاستقرار بشكل أسرع.

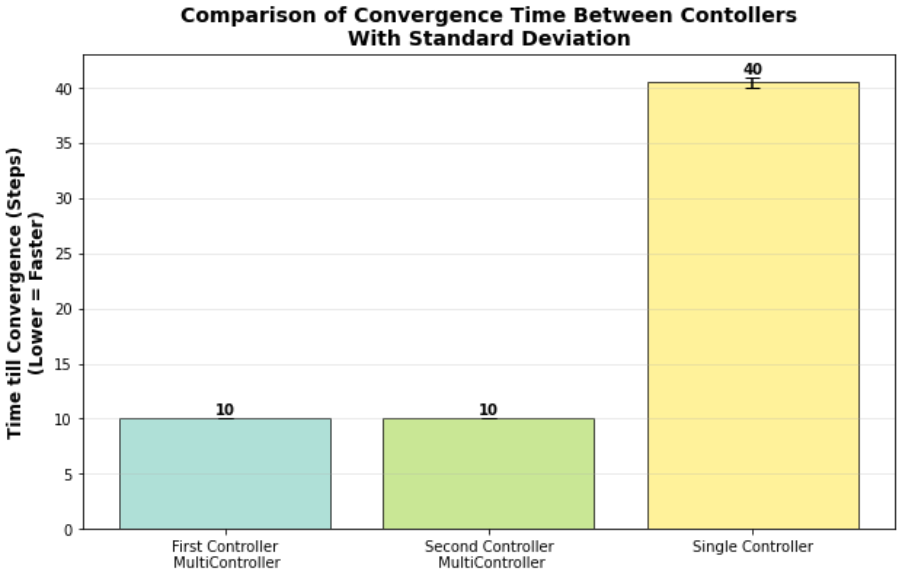
تخفيض العبء الحسابي:

– في الشبكة الأصلية:

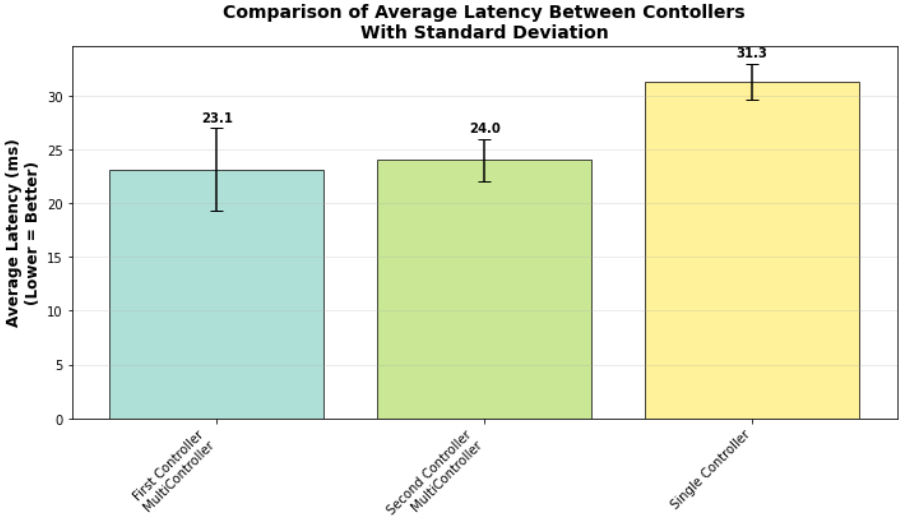
– المتحكم (`c0`) يتحمل عبءً كبيرًا لإدارة جميع العقد والتدفقات.

– في الشبكة المقسمة:

– كل متحكم يركز على نطاق صغير من العقد والتدفقات، مما يخفف العبء الحسابي على كل متحكم.




**الشكل 7: نتائج وقت التقارب لمتحكم واحد (الشكل 5) ومتحكمين (الشكل 6).**




**الشكل 8 : نتائج متوسط التأخير لمتحكم واحد (الشكل 5) ومتحكمين (الشكل 6).**

1. Single Controller: يشير إلى نظام يحتوي على متحكم واحد فقط، وهو المسؤول عن إدارة الشبكة بالكامل (شبكة غير مقسمة إلى نطاقات/دومينات).

2. First Controller: يشير إلى المتحكم الأول في نظام يحتوي على نطاقين (دومينين)، وهو المسؤول عن إدارة النطاق الأول.

3. Second Controller: يشير إلى المتحكم الثاني في نفس النظام، وهو المسؤول عن إدارة النطاق الثاني.

4. المعادلة الجديدة لحساب حجم q-table

عدد السويتشات لكل متحكم يصبح $\frac{N}{k}$

حجم الجدول Q لكل متحكم:

$$\underbrace{|Q(s,a)|}_{\text{(Size of 1 controller)}} = \underbrace{(\frac{N}{K})\hat{}M}_{\text{(Number of states)}} * \underbrace{\left(\left[\left(\frac{N}{K}\right)-1\right)\cdot M+1\right]}_{\text{(Number of actions)}}$$

الحجم الإجمالي للجداول Q:

$$\underbrace{|Q(s,a)|}_{\text{(total Size )}} = k. \underbrace{|Q(s,a)|}_{\text{(Size of 1 controller)}} \quad (8)$$

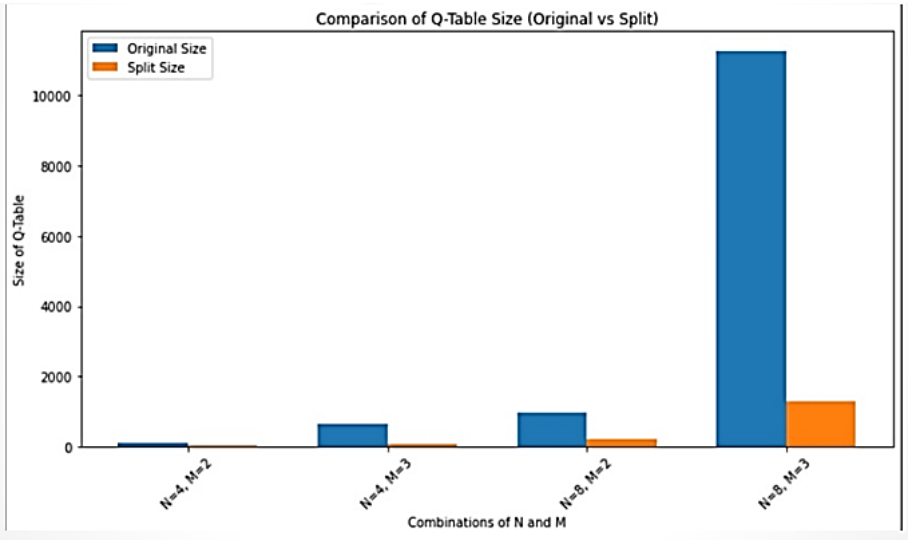
حيث ان :

N: عدد السويتشات الوسيطة .

M : عدد التدفقات.

K: عدد المتحكمات.

ومنه نصل الى النتائج التالية من اجل k=2:




**الشكل 8: حجم جدول Q في حال وجود متحكم واحد ومتحكمين كل واحد مسؤول عن مجال خاص به وذلك لقيم مختلفة لكل من N,M.**

**14.الاستنتاجات والأعمال المستقبلية:**

تم تقسيم الشبكة إلى دومينات لمعالجة مشكلة التوسع (`scalability`) المرتبطة بتزايد عدد التدفقات النشطة. في الشبكة الأصلية، كان المتحكم (`c0`) مسؤولاً عن إدارة جميع العقد (`s1`, `s2`, `s3`, `s4`, `s5`, `s6`)، مما أدى إلى مساحة حالة–فعل كبيرة ووقت تقارب طويل. بعد التقسيم إلى دومينين، أصبح كل متحكم يركز على نطاق صغير من العقد، مما خفض حجم المساحة الحالة–الفعل وزمن التقارب. النتائج أظهرت أن الشبكة المقسمة حققت زمن تقارب أسرع ومتوسط زمن وصول أقل مقارنة بالشبكة الأصلية.

استكشاف استراتيجيات ذكية أكثر لاستكشاف المسارات بهدف تعزيز قابلية التوسع. على سبيل المثال، قد تقتصر عملية استكشاف المسارات على الوصلات ذات الاستخدام العالي (التي تكون أكثر عُرضة للازدحام)، أو تُركّز فقط على التدفقات ذات المعدّل المرتفع، بينما يُوجَّه الجزء الأكبر من التدفقات ذات المعدّل المنخفض باستخدام توجيه تقليدي (مثل أقصر مسار)، أو باستخدام أساليب بديلة تُخفّف الضغط عن الوصلات المزدحمة (كالتوجيه عبر أطول المسارات مثلاً). كما أن العديد من التدفقات في الشبكات التشغيلية لا تتطلب زمن وصول قصير، بل تكتفي باتصال موثوق حتى لو تحمّلت تأخيرات طويلة. ويمكن في هذه الحالة توجيه هذه التدفقات المتسامحة مع التأخير (delay-tolerant flows) باستخدام خوارزميات توجيه تقليدية، مما يتيح اقتصار استخدام استراتيجيات الاستكشاف المعتمدة على التعلّم المعزز فقط على التدفقات الحساسة للزمن (low-latency flows).

في هذه الدراسة، تم تنفيذ آلية أولية لتبادل جداول التوجيه بين المتحكمات المتعددة عبر واجهة برمجة التطبيقات (REST API)، بهدف تمكين التنسيق البيني بين النطاقات المستقلة. ومع ذلك، لم يتم استغلال هذه الاتصالات بشكل كامل لتحسين قرارات التوجيه على مستوى الشبكة ككل. في العمل المستقبلي، يمكن تطوير بروتوكولات تعاون ذكية تتيح للمتحكمات مشاركة معلومات حيوية مثل: حالة الازدحام في الوصلات الحدودية، أو التنبؤات بسلوك التدفقات العابرة (inter-domain flows)، أو حتى سياسات Q-values المُتعلّمة ذات الصلة بالمسارات المشتركة. ومن خلال دمج هذه المعلومات في عملية اتخاذ القرار، يمكن لكل متحكم أن يُحسّن توجيه التدفقات العابرة لحدود نطاقه، ويقلل من التأخير الكلي وتجنب الازدحامات المتكررة عند نقاط التقاطع بين الأنظمة الذاتية (ASes). كما يمكن استغلال قناة الاتصال بين المتحكمات لتنفيذ استراتيجيات توجيه متناسقة، مثل توزيع التدفقات عالية المعدّل عبر مسارات متكاملة تمر بعدة نطاقات دون تعارض، أو تمكين استجابة

جماعيـــة للتغيـــرات المفاجئـــة فـــي الأحمـــال. وبهـــذا، يتحـــول النظـــام مـــن مجموعـــة مـــن المتحكمـــات المســـتقلة إلـــى شـــبكة متعاونـــة مـــن الـــوكلاء الـــذكيين، قـــادرة علـــى تحقيـــق أداء شمولي يفوق مجموع أداء أجزائها الفردية.

## 15. المراجع

1. Ahuja, Ravindra K., et al. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, 1993.
2. Maugendre, Marion. Development of a Performance Measurement Tool for SDN. 7 Oct. 2015.
3. Lantz, Bob, et al. Introduction to Mininet. Edited by Bob Lantz, 13 Feb. 2017.
4. Linkletter, Brian. "How to Use MiniEdit, Mininet’s Graphical User Interface." Brian Linkletter, 2 Apr. 2015.
5. Sharma, Shubham. Accurate Traffic Generation in the CheesePi Network. Master’s thesis, KTH Royal Institute of Technology, 2017.
6. Fang, Chao, et al. "Research on Routing Algorithm Based on Reinforcement Learning in SDN." Journal of Physics: Conference Series, vol. 1284, Aug. 2019, p. 012053.
7. Rischke, Julius, et al. "QR-SDN: Towards Reinforcement Learning States, Actions, and Rewards for Direct Flow Routing in Software-Defined Networks." IEEE Access, vol. 8, 2020, pp. 174773–174791.
8. Casas-Velasco, David M., et al. "Intelligent Routing Based on Reinforcement Learning for Software-Defined Networking." IEEE Transactions on Network and Service Management, vol. 18, no. 1, Mar. 2021, pp. 870–881.

9. Jalil, S. Q., et al. "DQR: Deep Q-Routing in Software Defined Networks." Proceedings of the International Joint Conference on Neural Networks (IJCNN), Jul. 2020, pp. 1–8.
10. Etengu, R., et al. "AI-Assisted Framework for Green-Routing and Load Balancing in Hybrid Software-Defined Networking: Proposal, Challenges and Future Perspective." IEEE Access, vol. 8, 2020, pp. 166384–166441.
11. Open Networking Foundation. Open Networking Foundation, 2014.
12. Bannour, Fetia, et al. "Distributed SDN Control: Survey, Taxonomy, and Challenges." IEEE Communications Surveys & Tutorials, vol. 20, no. 1, 2017, pp. 333–354.
13. Benzekki, Kamal, et al. "Software-Defined Networking (SDN): A Survey." Security and Communication Networks, vol. 9, no. 18, 2016, pp. 5803–5833.
14. Waseem, Q., et al. "Software-Defined Network (SDN) Forensic." Symmetry, vol. 13, no. 5, 2021, pp. 767–785.
15. Sutton, Richard S., and Andrew G. Barto. Reinforcement Learning: An Introduction. 2nd ed., MIT Press, 2018.
16. Xie, Junfeng, et al. "A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges." IEEE Communications Surveys & Tutorials, vol. 21, no. 1, 2018, pp. 393–430.
17. Jang, Beomseok, et al. "Q-Learning Algorithms: A Comprehensive Classification and Applications." IEEE Access, vol. 7, 2019, pp. 133653–133667.
18. Francois-Lavet, Vincent, et al. "An Introduction to Deep Reinforcement Learning." Foundations and Trends in Machine Learning, vol. 11, nos. 3–4, 2018, pp. 219–354.
19. Sachs, J., et al. "Adaptive 5G Low-Latency Communication for Tactile Internet Services." Proceedings of the IEEE, vol. 107, no. 2, Feb. 2019, pp. 325–349.

20.Xiang, Z., et al. "Reducing Latency in Virtual Machines: Enabling Tactile Internet for Human-Machine Co-Working." IEEE Journal on Selected Areas in Communications, vol. 37, no. 5, May 2019, pp. 1098–1116.
21.Lin, Shih-Chun, et al. "QoS-Aware Adaptive Routing in Multi-Layer Hierarchical Software Defined Networks: A Reinforcement Learning Approach." Proceedings of the IEEE International Conference on Services Computing (SCC), Jun. 2016, pp. 25–33.
22.Montazerolghaem, Ahmadreza, and Somaye Imanpour. "Evaluation and Performance Analysis of the Ryu Controller in Various Network Scenarios." arXiv, 25 May 2025, arxiv.org/abs/2505.19290.
23."Performance Evaluation of Ryu Controller in Software Defined Networks." Journal of Al-Qadisiyah for Computer Science and Mathematics, vol. 14, no. 1, 2022, pp. 1–7, doi.org/10.29304/jqcm.2022.14.1.879.
24.Chauhan, Pinkey, and Mithilesh Atulkar. "Ryu Controller-Based Attack Detection and Mitigation Method in Software Defined Internet of Things." International Journal of Engineering Trends and Technology, vol. 71, no. 9, 2023, pp. 138–156, doi.org/10.14445/22315381/IJETT-V71I9P213.
25."POX and RYU Controller Performance Analysis on Software Defined Networks." EAI Endorsed Transactions on Internet of Things, vol. 9, no. 1, 2023, publications.eai.eu/index.php/IoT/article/view/2821.
26.Ashok, R., et al. "iPerf—The Ultimate Speed Test Tool for TCP, UDP and SCTP." Journal of Computer and Communications, vol. 8, no. 9, 2020, pp. 11–19, www.scirp.org/journal/paperinformation?paperid=1026461 .
27.NetBeez. "iPerf and Network Performance Testing." NetBeez Blog, 2023, netbeez.net/blog/how-to-use-iperf/.

28.Oracle Corporation. "Use iPerf to Test the Throughput Inside an OCI Hub and Spoke VCN Routing Architecture." Oracle Documentation, 2023, docs.oracle.com/en/learn/iperf-testing-in-oci/index.html.
29.Zamfir, Radu, et al. "Mobile Network Operators' Assessment Based on Drive-Test Scenarios Using Open-Source Tools." Applied Sciences, vol. 14, no. 3, 2024, p. 1268, doi.org/10.3390/app14031268.

# تصميم نظام كشف لحظي لهجمات DDoS في SDN بالاعتماد على برمجة مستوى البيانات بلغة P4 ونموذج XGBoost

م. محسن ثابت احمد[1] د.م. مياد جابر[2]

**الملخص**

يهدف هذا البحث إلى تقديم إطارمتكامل يجمع بين لغة برمجة مستوى البيانات P4 ومتحكم Ryuونموذج الذكاء الاصطناعي (XGBoost)، للكشف عن الهجمات في الشبكات المعرفة برمجياً (SDN) وذلك في الزمن الحقيقي. يعتمد النظام على استخراج ميزات التدفق مباشرة داخل مستوى البيانات، ثم إرسالها إلى المتحكم حيث يتم تحليلها باستخدام نموذج ذكاء اصطناعي مدرّب مسبقاً. ولزيادة دقة الكشف، تم اعتماد عتبة التكيّف (Adaptive Thresholding) التي تسمح بتعديل العتبة حسب الأداء اللحظي. أظهر النظام المقترح تفوقاً ملحوظ من حيث زمن الكشف، إذ لم يتجاوز 0.3 ثانية، مما يعكس سرعة استجابة عالية للنظام. كما تجاوزت قيم مؤشرات الأداء، بما في ذلك الدقة (Accuracy)، معدل F1، الدقة النوعية (Precision) و الاسترجاع (Recall) نسبة تتجاوز 99.9% ، مما يجعله مناسباً للتطبيق في البيئات الحرجة.

**الكلمات المفتاحيّة:** الشبكات المعرّفة برمجياً، التعلم الآلي، لغة p4، هجمات حجب الخدمة الموزعة.

*1*-مهندس في قسم النظم والشبكات الحاسوبية-كلية الهندسة المعلوماتية-جامعة دمشق.

*2*- مدرس في قسم النظم والشبكات الحاسوبية-كلية الهندسة المعلوماتية-جامعة دمشق.

## Design of a Real-Time DDoS Detection Framework in SDN Based on P4 Data-Plane Programming and XGBoost


### Abstract

This research introduces an integrated framework that unites the data plane programming language P4, the Ryu controller, and an artificial intelligence model (XGBoost) for real-time attack detection in Software-Defined Networks (SDNs). The system captures and extracts flow features directly within the data plane, forwarding them to the controller for analysis by a pre-trained AI model. To improve detection accuracy, an Adaptive Thresholding mechanism dynamically adjusts thresholds based on live performance. The proposed approach achieved exceptional detection times, consistently under 0.3 seconds, demonstrating high responsiveness. Moreover, performance metrics, including Accuracy, F1-score, Precision, and Recall, surpassed 99.9%, highlighting the system's suitability for critical environments.


**Keywords:** Software-Defined Networks, Machine Learning, P4 Language, Distributed Denial of Service (DDoS) Attacks**.**

## 1. مقدمة:

شهدت شبكات الحاسوب تطوراً جذرياً خلال السنوات الأخيرة، إذ انتقلت من البنى التقليدية المعتمدة على أجهزة متخصصة ومرتبطة ببروتوكولات ثابتة، إلى نماذج أكثر مرونة وقابلية للبرمجة. تُعد الشبكات المعرفة برمجياً (Software-Defined Networking – SDN) أبرز هذه النماذج الحديثة، حيث تفصل بين مستوى التحكم (Control Plane) المسؤول عن اتخاذ القرارات، ومستوى البيانات (Data Plane) المعني بتمرير الحزم. يوفّر هذا الفصل ميزة الإدارة المركزية للشبكة، مع رؤية شاملة لحركة المرور، وقدرة على ضبط السياسات وتنفيذها ديناميكياً بما يعزز مرونة الشبكة ويُسهِّل تطويرها مقارنة بالشبكات التقليدية التي تعتمد على أجهزة مغلقة محدودة القدرات[1].

رغم ما تقدمه SDN من قدرات واعدة، إلا أن هذا النموذج يفتح الباب أمام تحديات أمنية خطيرة ناتجة عن مركزية التحكم، إذ تصبح هدفاً مباشراً للهجمات الإلكترونية. تمثل هجمات حجب الخدمة الموزّعة (DDoS) أحد أبرز التهديدات التي تستغل طبيعة SDN المعتمدة على تبادل مستمر للرسائل بين المتحكم والمبدلات. يؤدي هذا النوع من الهجمات إلى إغراق المتحكم بالطلبات أو استنزاف موارد المبدلات، مما يعرقل استجابة الشبكة وقد يتسبب في انهيار جزئي أو كامل لمنظومتها[2]. من هنا، تتجلى مشكلة البحث في غياب إطار أمني متكامل يعمل ذاتياً في شبكات SDN، قادر على الكشف المبكر والتكيف مع أنماط هجوم متنوعة ومعقدة والاستجابة اللحظية ، بالإضافة الى الموازنة بين الأمن والأداء بحيث لا يترتب على آليات الحماية استنزاف موارد المعالجة أو تراجع جودة الخدمة. تعتمد الاتجاهات الحديثة في هذا المجال على دمج الذكاء الاصطناعي مع بنى SDN لتحقيق قدر أكبر من التحليل والتكيّف مع أنماط حركة المرور المتغيرة[3,4,5,6,7,8]. يُعد التعلم الآلي من أكثر الأساليب فعالية في تمييز السلوك الشاذ المرتبط بهجمات DDoS. كما توفر تقنيات برمجة مستوى البيانات مثل P4 آلية متقدمة لاستخراج ميزات

التدفق مباشرةً من المبدل بدقة وزمن استجابة منخفض، وهو ما يحقق تكاملاً عملياً بين قدرات الذكاء الاصطناعي وأتمتة معالجة الحزم.

بناءً على ذلك، يهدف هذا البحث إلى تطوير نظام أمني متكامل للكشف اللحظي عن هجمات DDoSفي بيئات SDN، من خلال دمج برمجة مستوى البيانات باستخدام P4 مع نموذج XGBoostداخل متحكمRyu ، إضافةً إلى اعتماد آلية العتبة التكيفية Adaptive) (Thresholdingللتكيّف مع التغيرات المستمرة في حركة مرور الشبكة وتقليل الإنذارات الكاذبة. يسعى هذا النظام إلى تحقيق توازن بين الدقة العالية والسرعة وقابلية التطبيق الفعلي، بما يجعله مناسباً للبيئات الحرجة التي تتطلب حماية مستمرة وفعالة.

## 2. الدراسات السابقة

اقترحت الدراسة [3] نظاماً ذكياً مدمجاً داخل متحكم SDN، يعتمد على تحليل حركة المرور في الزمن الحقيقي باستخدام محاكاة EstiNet، بالاستناد إلى مجموعتي البيانات DDoS-SDN و InSDN.أظهر النظام تفوق على النماذج التقليدية مثل Decision Tree و Random Forest في التصنيف الثنائي (F1 = 1)، بفضل تطبيق هيكل تصنيف هرمي متعدد الفئات (HMC)لمعالجة مشكلة عدم توازن البيانات. بينما ركزت الدراسة [4] على تصميم نظام معماري مرن وقابل للتوسع للكشف عن هجمات DDoS في طبقتي النقل والتطبيق، باستخدام مجموعتي البيانات CICDDoS2019 و CICDoS2017. أظهرت النتائج تفوق نماذج GRU و LSTM بدقة تجاوزت 99% ومعدل اكتشاف يفوق 95%، مع كفاءة عالية في استهلاك الموارد وسرعة المعالجة ضمن بيئة محاكاة واقعية. وفي سياق مماثل، قدّمت الدراسة[5] نموذجاً هجينياً يجمع بين أسلوب الإنتروبيا للكشف المبكر خلال أول 250 حزمة ضارة، ونماذج التعلم العميق (LSTM

، GRU، RNN، MLP) لتصنيف أنواع الهجمات، محققاً دقة بلغت 99.42%. أما الدراسة [6] فقد قارنت بين نماذج التعلم العميق (LSTM، CNN) والخوارزميات التقليدية KNN، SVM و Naive Bayes ضمن بيئة محاكاة Mininet، حيث حقق نموذج KNN أعلى دقة (99.4%) ، في حين أظهر LSTM أفضل توازن بين الدقة والاستدعاء، مما يجعله أكثر موثوقية في الكشف شبه الحقيقي. وفي إطار تحسين الدقة والتعميم، قدّمت الدراسة[7] نموذجاً هجينياً مبتكراً يُسمى DDosTCيعتمد على دمج Transformer مع CNN باستخدام مجموعة بيانات CICDDoS2019، محققاً دقة 99.82% متفوقاً على نماذج LSTM و GRU وCNN، رغم محدودية التقييم من حيث التكلفة الحسابية والاختبار العملي. أخيراً، عرضت الدراسة [8] إطار عمل دفاعي شامل يُعرف باسم CC-Guard لحماية وحدات تحكم SDN متعددة النطاقات، من خلال بنية تعتمد على كشف ثنائي المراحل: المرحلة الأولى تستخدم إنتروبيا عناوين IP (عتبات: المصدر 2.5، الوجهة 1.0) للكشف المبدئي، بينما تعتمد المرحلة الثانية على نموذج CNN-GRU-Attention لتحليل 24 خاصية من جداول التدفق، محققاً دقة 0.999% كما استخدم النظام خوارزمية جينية لترحيل المبدلات وتخفيف الضغط عن المتحكمات، مع قدرة على استعادة الحالة الطبيعية خلال 8–10 ثوانٍ. أظهرت الدراسات السابقة تنوعاً ملحوظاً في الأساليب المستخدمة لرصد هجمات حجب الخدمة الموزعة (DDoS)، حيث ركزت على تحسين الدقة و زمن الكشف باستخدام نماذج ذكاء اصطناعي عميقة و متطورة، وعالجت مشكلة عدم توازن البيانات.

أظهرت الدراسات السابقة تنوعاً واسعاً في الأساليب المستخدمة للكشف عن هجمات حجب الخدمة الموزّعة (DDoS) في الشبكات المعرفة برمجياً، حيث اعتمدت بعض الأعمال على نماذج تعلم آلي داخل المتحكم مثل Decision Tree و Random Forest وHMC، بينما ركزت دراسات أخرى على نماذج التعلم العميق مثل LSTM و GRU و CNN لتحقيق أعلى دقة في التصنيف، كما تبنت أبحاث إضافية أساليب هجينة تعتمد على الإنتروبي أو دمج الشبكات العصبية مع

المحولات (Transformers)، في حين تناولت بعض الدراسات أنظمة دفاع متعددة المراحل أو هياكل حماية للمتحكمات متعددة النطاقات. رغم أهمية هذه الجهود، إلا أنها تشترك في قيود بارزة تتمثل في اعتمادها شبه الكامل على مستوى التحكم لاستخراج ميزات التدفق وتحليلها، مما يزيد العبء على المتحكم ويرفع زمن الكشف، إضافة إلى استخدام عتبات ثابتة تحدّ من التكيف مع حركة المرور المتغيرة، وعدم الاستفادة من قدرات برمجة مستوى البيانات (P4) في استخراج الميزات لحظياً.

انطلاقاً من تحليل هذه الأدبيات، تتضح فجوة بحثية مركزية تتمثل في غياب دراسة تدمج بين استخراج ميزات التدفق مباشرةً في مستوى البيانات باستخدام P4 وبين نموذج ذكاء اصطناعي عالي الأداء داخل المتحكم، مع تطبيق آلية عتبة ديناميكية تتيح كشفاً لحظياً أكثر دقة وتخفف العبء عن المتحكم، إضافة إلى نقص التقييم في سيناريوهات زمن حقيقي واقعية. من هنا، تبرز الحاجة إلى إطار متكامل يوازن بين سرعة استخراج الميزات على مستوى البيانات، والدقة العالية في التصنيف، والقدرة على التكيف مع حركة المرور عبر Adaptive Thresholding، وتقليل الضغط على المتحكم، وهي المتطلبات التي يسعى هذا البحث إلى تحقيقها عبر نظام يجمع بين P4 ، Ryu، XGBoost، و آلية العتبة التكيفية للكشف اللحظي الدقيق عن هجمات DDoS في SDN.

### 3. طرائق البحث وموادها

تمت محاكاة النظام المقترح باستخدام بيئة Mininet، حيث تم تطوير أكواد المتحكم بلغة Pythonوبرمجة المبدل باستخدام P4. أجريت عمليات تدريب نموذج XGBoostعلى منصة Google Colabللاستفادة من موارد الحوسبة العالية، في حين جرى اختبار النظام عمليًا على جهاز يعمل بنظام Ubuntu 20.04، مزود بمعالج Intel Core i5 من الجيل العاشر وذاكرة عشوائية بسعة 8 جيجابايت. لعبت خصائص الجهاز دوراً مهماً في الاختبار، حيث أثرت سرعة المعالج على قدرة النظام في معالجة تدفقات البيانات بسرعة، وأتاحت سعة الذاكرة المستخدمة

إدارة بيئة المحاكاة وتشغيل النموذج دون تأخير كبير، مما مكّن من تقييم زمن الكشف ودقة النظام بشكل فعّال يعكس الأداء المتوقع في الظروف الواقعية.

### 1.3. الإطار النظري

يستند هذا البحث إلى مجموعة من المفاهيم والتقنيات الأساسية التي تشكل مجتمعةً الهيكل النظري للنظام المقترح. يمكن تلخيص هذه الركائز على النحو التالي:

1. الشبكات المعرّفة برمجياً (SDN): هي نموذج معماري للشبكات يفصل بين مستوى التحكم (Control Plane) المسؤول عن اتخاذ قرارات التوجيه والسياسات، ومستوى البيانات (Data Plane) المسؤول عن تمرير الحزم وفقاً لتلك القرارات[1]. يتميز هذا النموذج بمركزيّة الإدارة من خلال متحكم (Controller) يمتلك رؤية شاملة (Global View) للشبكة، مما يمنحه مرونة عالية في إدارتها وبرمجتها ديناميكياً لمواجهة متطلبات متغيرة، بما في ذلك التهديدات الأمنية[2].

2. برمجة مستوى البيانات بلغة P4: هي لغة برمجة مفتوحة المصدر تُستخدم لبرمجة سلوك مستوى البيانات في أجهزة الشبكات (كالمبدلات) بشكل مستقل عن البروتوكولات. تسمح P4بتعريف كيفية معالجة الحزم بدءاً من رؤوس البروتوكولات وحتى العمليات الحسابية المعقدة، مما يمكّن من استخراج ميزات مخصصة لتحليل حركة المرور (مثل إحصائيات التدفق) مباشرةً داخل المبدل [13]، وهو ما يُعد حجر الأساس في نظام الكشف المقترح.

3. التعلم الآلي والكشف عن الهجمات: يُعد التعلم الآلي (Machine Learning) أحد فروع الذكاء الاصطناعي الذي يسمح للأنظمة بالتعلم من البيانات دون أن تتم برمجتها بشكل صريح. تُستخدم خوارزميات التعلم الآلي لبناء نماذج قادرة على تمييز الأنماط الطبيعية لحركة المرور عن تلك الضارة في مجال أمن الشبكات. يعتمد هذا البحث بشكل خاص على خوارزمية XGBoost ، وذلك استناداً إلى مبررات أدائية. تُعرف XGBoost بأنها

خوارزمية تعزيز تدرجي (Gradient Boosting) متقدمة، تتميز بكفاءتها العالية في معالجة البيانات المنطقية والرقمية، وقدرتها على تمييز الأنماط غير الخطية في بيانات حركة المرور الشبكية، والتي تعد سمة أساسية في تمييز الهجمات عن الحركة العادية. تكمن قوة XGBoost في آليتها التي تجمع بين عدد كبير من أشجار القرار الضعيفة (Weak Learners) في نموذج ensemble قوي، مع تضمين مصطلحات تنظيم (Regularization) للحد من ظاهرة الإفراط في التخصيص (Overfitting)، مما يضمن تعميماً أفضل على البيانات غير المرئية [10]. أظهر XGBoost تفوقاً ملحوظاً في العديد من مهام التصنيف الثنائي بالمقارنة مع خوارزميات التعلم الآلي التقليدية الأخرى ، لا سيما في مجال أمن الشبكات، حيث تفوقت على خوارزميات مثل Random Forest و SVMو KNN من حيث الدقة والسرعة [15]، وهو عامل حاسم في أنظمة الكشف اللحظي.

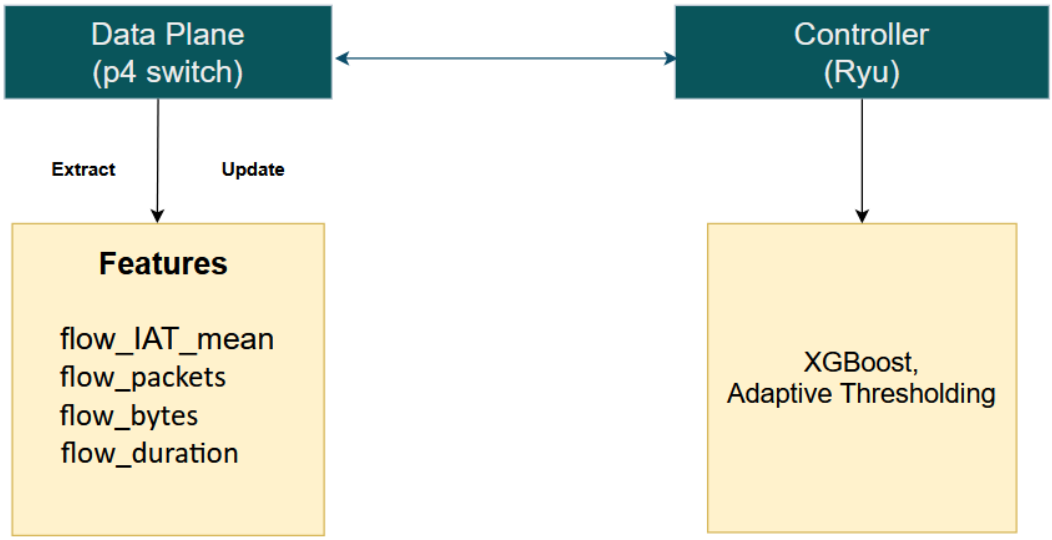
4. المتحكم: معظم الدراسات التي تعتمد على المتحكمات التقليدية تستخدم منصات ثقيلة مثل ONOSو OpenDaylight، والتي تُعد أقل مرونة في دمج خوارزميات التعلم الآلي وصعبة التخصيص مقارنة بمتحكم Ryu، الذي يتميز بخفة وزنه وسهولة برمجته بلغة Python[14]، مما يجعله مناسباً لدمج نماذج مثل XGBoost ومعالجة بيانات التدفق لحظياً بكفاءة أكبر.

5. آلية العتبة التكيفية (Adaptive Thresholding): العتبة (Threshold) هي القيمة التي يتم مقارنة ناتج نموذج التصنيف بها لتحديد إذا ما كان التدفق هجوماً أم لا. بدلاً من استخدام عتبة ثابتة قد لا تكون مناسبة لجميع ظروف حركة المرور، تقوم آلية العتبة التكيفية بتعديل هذه القيمة تلقائياً بناءً على الأداء اللحظي للنظام وطبيعة حركة المرور المتغيرة، مما يساهم في تحسين دقة الكشف وتقليل نسبة الإنذارات الكاذبة (False Positives) [4,7].

يشكل تكامل هذه المكونات الأربعة (P4,RYU, XGBoost, Adaptive Thresholding) الإطار النظري المتين الذي يبني عليه هذا البحث نظامه المتكامل للكشف عن الهجمات والاستجابة لها في بيئات SDN بفعالية وسرعة عاليتين.

### 2.3. الحل المقترح

يعتمد الحل المقترح على بناء نظام مكوّن من وحدتين أساسيتين: وحدة البيانات و وحدة التحكم. يوضح الشكل (1) البنية العامة للنظام المقترح، حيث تقوم وحدة البيانات بجمع وتحليل حركة المرور الواردة من الشبكة، بينما تتولى وحدة التحكم عملية الكشف، واتخاذ القرارات الخاصة بالتخفيف من الهجمات بالاعتماد على نموذج XGBoost و آلية العتبة التكيفية.




الشكل (1) البنية العامة للنظام المقترح

### 1.2.3. وحدة البيانات (P4 Switch)

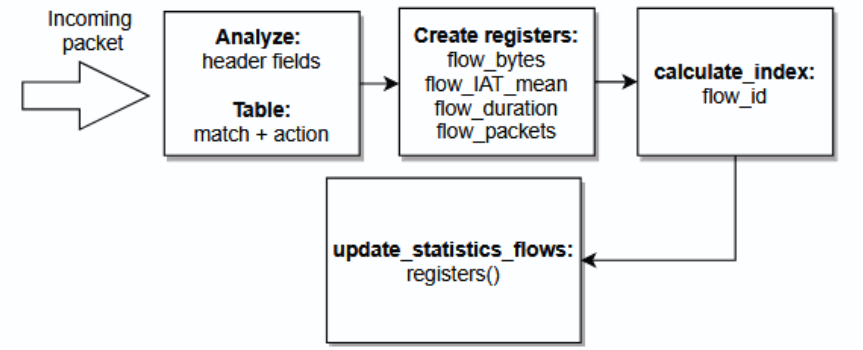
تقوم وحدة البيانات في النظام المقترح بعدة مهام رئيسية بناءً على مفاهيم سابقة[9] كما في الشكل (2)، تشمل هذه المهام:

- تعريف رؤوس البروتوكولات (Ethernet, IPv4, TCP).
- حساب ميزات التدفق: عدد الحزم، حجم البايتات، متوسطIAT ، مدة التدفق.
- تمرير النتائج إلى المتحكم عبر P4Runtime/Thrift.

يمكن التعبير عنها في كود P4 التالي:

```
header ethernet_t { ... }
header ipv4_t { ... }
action compute_flow_stats() {
    meta.packet_count = meta.packet_count + 1;
    meta.byte_count = meta.byte_count +
packet_length;
    meta.iat = now - meta.last_packet_time;
    meta.last_packet_time = now;
}
control egress(inout headers hdr, inout
metadata meta) {
    apply {
        if (meta.packet_count % sampling_rate
== 0) {
            send_to_controller(meta);
        }
```

```
        }
    }
```




الشكل (2) آلية عمل وحدة البيانات [9]

يوضح الجدول (1) أنواع السجلات التي تخرجها وحدة البيانات وترسلها للمتحكم:

جدول (1) أنواع السجلات التي تخرج من وحدة البيانات

| اسم السجل في P4 | الوصف |
|---|---|
| flow_bytes_reg | سجل إجمالي البايتات في التدفق |
| flow_packets_reg | سجل إجمالي الحزم في التدفق |
| flow_duration_reg | سجل مدة التدفق |
| flow_total_IAT_reg | سجل مجموع الفترات بين الحزم (IAT) |

### 2.2.3. وحدة التحكم (Ryu Controller)

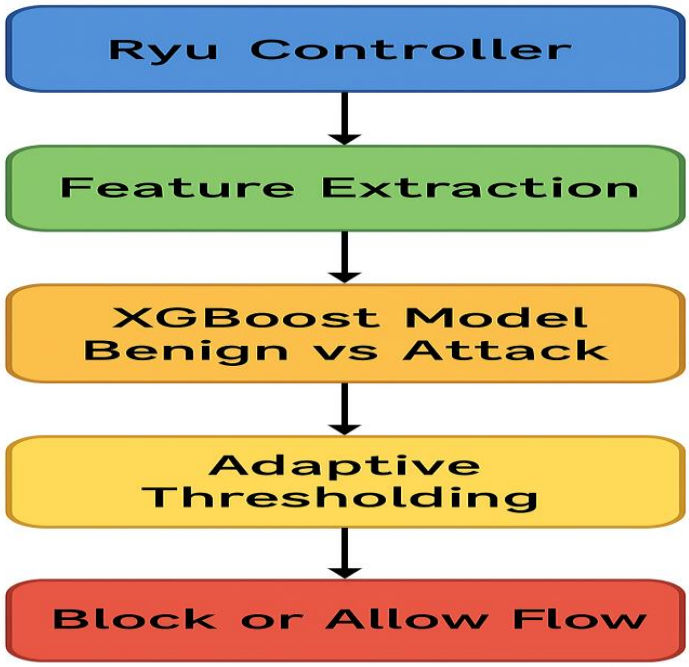
تتولى وحدة التحكم في النظام المقترح إدارة عمليات الكشف واتخاذ القرار، من خلال:

- بناء تطبيق RyuApp بلغة Python.
- تحميل نموذج XGBoost مدرّب مسبقاً Scaler.
- استقبال إحصاءات التدفق (السجلات) من switch
- استخراج الخصائص وبناء متجه الميزات (feature vector).
- تطبيق التصنيف (Benign vs Attack) باستخدام النموذج.
- استخدام Adaptive Thresholding لتعديل الحد الفاصل حسب أداء النظام.
- إعادة كتابة قرارات إلى الـ switch عبر جدول ml_results أو قواعد drop.

يكمن التعبير عن عمل وحدة التحكم في كود البايثون التالي والشكل (3):

```
class DDoSDetector(app_manager.RyuApp):
def __init__(self, *args, **kwargs):
super(DDoSDetector, self).__init__(*args,
**kwargs)
self.model =
pickle.load(open('xgboost_model.pkl', 'rb'))
self.scaler = pickle.load(open('scaler.pkl',
'rb'))
@set_ev_cls(ofp_event.EventOFPPacketIn,
MAIN_DISPATCHER)
self.adaptive_threshold = 0.5
self.target_fpr = 0.01
```

```
self.delta = 0.01
self.window_size = 100
self.decision_history = []
self.fpr_history = []
self.MIN_THRESHOLD = 0.3
self.MAX_THRESHOLD = 0.99
def packet_in_handler(self, ev):
features = self.extract_features(ev)
features_scaled =
self.scaler.transform([features])
prediction = self.model.predict(features_scaled)
probability=self.model.predict_proba(features_sca
led)
threshold = self.compute_adaptive_threshold()
if probability[0][1] > threshold:
self.block_flow(ev) إجراء الصد
def compute_adaptive_threshold(self):
return adaptive_threshold
```




الشكل (3) آلية عمل وحدة التحكم

### 3.2.3. الميزات المستخرجة وطريقة تجميع البيانات

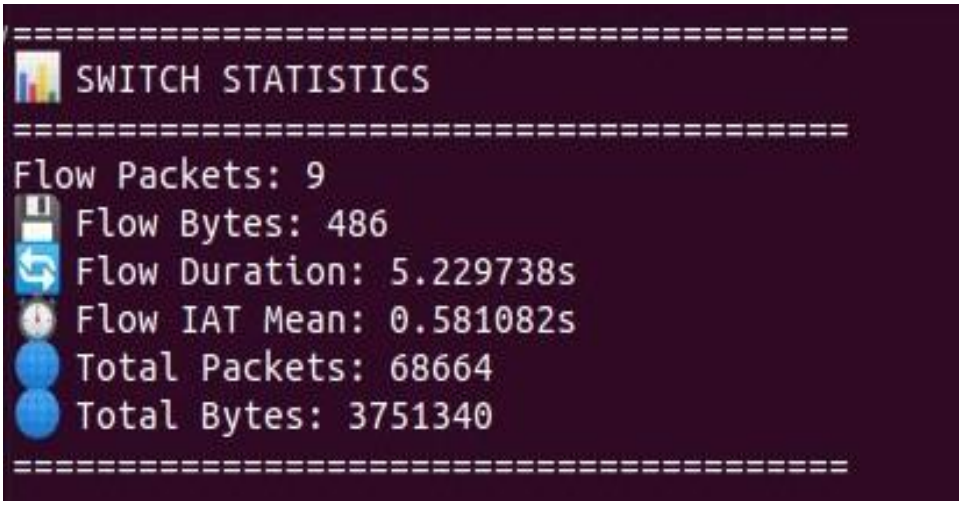
يعتمد النظام على مجموعة من الميزات الإحصائية الحيوية المستخرجة ديناميكياً من خلال وحدة البيانات (مبدل P4 ) لتمكين المتحكم من اتخاذ قرار التصنيف (هجوم أم حركة عادية). تم اختيار هذه الميزات لقدرتها على عكس السلوك الشاذ لحركة مرور هجمات DDoS[9]، وهي:

- إجمالي الحزم في التدفق (flow_packets_reg): العدد الإجمالي للحزم ضمن نافذة زمنية أو تدفق معين.
- إجمالي البايتات في التدفق (flow_bytes_reg): الحجم الكلي للبيانات ضمن نفس التدفق.

- مدة التدفق (flow_duration_reg): الفترة الزمنية الكلية لنشاط التدفق.

- مجموع الفترات بين الحزم (flow_total_IAT_reg): المجموع التراكمي للفترات الزمنية بين الحزم المتتالية. يتم تحديث هذه السجلات مباشرة داخل مبدل P4 مع مرور كل حزمة. لتجنب إرباك وحدة التحكم، يتم اعتماد آلية أخذ العينات (Sampling)، حيث يتم إرسال ملخص هذه الإحصائيات إلى متحكم Ryu.

يبين الشكل (4) عينة للبيانات المجمعة التي يستقبلها المتحكم من المبدل أثناء محاكاة هجوم DDoS.




الشكل (4) عينة من البيانات التي تجمعها وحدة البيانات ويستقبلها المتحكم

### 4.2.3. آلية عمل العتبة التكيفية (Adaptive Thresholding)

تم دمج آلية العتبة التكيفية ضمن وحدة التحكم لتعزيز دقة النظام في التمييز بين حركة المرور الطبيعية وهجمات حجب الخدمة الموزعة (DDoS) بشكل ديناميكي. تختلف هذه الآلية جوهرياً عن أنظمة الكشف التقليدية التي تعتمد على عتبة ثابتة (Static Threshold)، حيث أنها قادرة

على تعديل قيمة العتبة المستخدمة لاتخاذ قرار التصنيف تلقائياً بناءً على الأداء اللحظي للنظام وطبيعة حركة المرور المتغيرة. الهدف الأساسي هو تحقيق توازن مثالي بين تقليل الإنذارات الكاذبة (False Positives) الناتجة عن تصنيف الحركة الشرعية على أنها هجوم، وتقليل (False Negatives)التي تفوت على النظام اكتشاف هجمات خبيثة قد تكون خفية.

تعمل العتبة التكيفية في متحكم Ryu كالتالي:

1. **تحديد العتبة الابتدائية:** يتم البدء بعتبة أولية (على سبيل المثال، 0.5) يتم ضبطها بناءً على التحليل الأولي لأداء النموذج على بيانات التدريب.

2. **مراقبة الأداء اللحظي:** يقوم النظام بمراقبة وتخزين نتائج التصنيف خلال نافذة زمنية متحركة (Sliding Window) تشمل أحدث عدد من التدفقات (مثلاً، آخر 100 تصنيف).

3. **حساب معدل الإنذارات الكاذبة (FPR):** يتم حساب معدل الإنذارات الكاذبة الحالي (current_fpr) ضمن هذه النافذة المتحركة. يمثل هذا المعدل نسبة التدفقات الطبيعية التي تم تصنيفها خطأً على أنها هجوم.

4. **مقارنة الأداء بالهدف:** تتم مقارنة current_fpr بمعدل الإنذارات الكاذبة المستهدف (target_fpr)، لضمان جودة الأداء (مثلاً، 1% أو 0.01).

5. **تعديل العتبة ديناميكياً:** بناءً على نتيجة المقارنة:

- **إذا كان current_fpr > target_fpr:** يشير هذا إلى وجود عدد كبير جداً من الإنذارات الكاذبة. لذلك، يتم زيادة قيمة العتبة الحالية بمقدار ثابت (delta، مثلاً 0.01) لجعل عملية التصنيف أكثر تشدداً، مما يتطلب ثقة أعلى من النموذج ([1][0]probability) لاعتبار التدفق هجوماً.

- **إذا كان current_fpr <= target_fpr:** يشير هذا إلى أن النظام دقيق بما فيه الكفاية. في هذه الحالة، يمكن خفض قيمة العتبة بشكل طفيف (مثلاً، delta / 2) لزيادة حساسية النظام لاكتشاف الهجمات الخفية، مع الحفاظ على مستوى مقبول من الإنذارات الكاذبة.

**6. الحدود الآمنة للعتبة:** يتم تقييد القيمة الجديدة بين حد أدنى (مثلاً، 0.5) وحد أقصى (مثلاً، 0.99) لضمان بقاء العتبة في نطاق منطقي وفعال.

يتكامل نموذج التصنيف القائم على خوارزمية XGBoost وآلية العتبة التكيفية، حيث تُكمّل كل منهما الأخرى لتحقيق كفاءة تشغيلية عالية. بينما يقوم نموذج الذكاء الاصطناعي بدور "المحلل الخبير" الذي يقدّر احتمالية أن يكون التدفق هجوماً بناءً على الميزات الإحصائية المستخرجة، تأتي آلية العتبة التكيفية كـ "منظّم ذكي" يترجم هذه الاحتمالات إلى قرارات عملية. فعلياً، ينتج نموذج XGBoost درجة احتمالية بين ( 0 و 1) تعبّر عن ثقته في تصنيف التدفق كهجوم. هنا تتدخل العتبة التكيفية كمعيار حيوي ومتغير حيث إذا تجاوزت الدرجة الاحتمالية قيمة العتبة الحالية، يُصنّف التدفق كهجوم. ولا تقتصر العلاقة عند هذا الحد، بل تمتد إلى التكيف الديناميكي حيث تراقب الآلية أداء النموذج في البيئة التشغيلية الفعلية (معدل الإنذارات الكاذبة) وتُعدّل قيمة العتبة تلقائياً صعوداً أو هبوطاً لتحقيق التوازن الأمثل بين الحساسية العالية للهجمات وتقليل الإيجابيات الكاذبة.

يحوّل هذا التكامل النظام من كاشف سلبي يعتمد على تنبؤات ثابتة إلى نظام استباقي ذكي، قادر على تحسين دقته ذاتياً والاستجابة بمرونة للظروف المتغيرة لحركة المرور الشبكية. تم تصميم العتبة التكيفية والنظام ككل ليكون مُحسّناً لاكتشاف هجمات حجب الخدمة الموزعة (DDoS) التي تتميز بتغيرات كبيرة في إحصائيات التدفق الشبكي.

### 3.3.آلية تدريب الخوارزمية:

تم اختيار مجموعة البيانات DrDoS_UDP.csv و Syn.csv من CICDDoS2019 [11]. تم تدريب النموذج باستخدام خوارزمية XGBoost مع ضبط المعاملات عبر RandomizedSearchCVكالتالي:

- learning_rate = 0.1
- max_depth = 6
- n_estimators = 200
- subsample = 0.9
- colsample_bytree = 0.8
- scale_pos_weight = 1
- eval_metric = 'logloss'
- objective = 'binary:logistic'

تم استخدام F1-score كمؤشر رئيسي للتقييم، كما تم تحديد العتبة المثلى للتصنيف استناداً إلى إحصائيةYouden لتحسين دقة القرار. تم التركيز على ميزات أساسية تشمل flow_duration ، flow_IAT_mean، flow_bytes، و flow_packets التي تعبر عن خصائص التدفق الشبكي ذات الصلة باكتشاف الهجمات.

جرى تقسيم البيانات إلى 70% للتدريب و30% للاختبار، مع تطبيق StandardScaler لتطبيع القيم و SMOTE لموازنة الفئات ومعالجة مشكلة عدم توازن البيانات. أُجري تقييم شامل للأداء باستخدام مقاييس متعددة تشمل الدقة (Accuracy)، الاستدعاء (Recall)، الدقة التصنيفية(Precision)، F1-Score و ROC-AUC، إضافةً إلى تحليل أهمية الميزات لتحديد أكثر الخصائص تأثيراً في عملية الاكتشاف. وأخيراً، تم حفظ النموذج وجميع الملفات المساعدة مثل (scaler و label_encoder).

أظهرت نتائج تدريب الخوارزمية المستخدمة أداءً عالياً ونتائج متميزة على مختلف مقاييس التقييم، حيث بلغت دقة النموذج (Accuracy) 0.9981، بينما سجلت نسبة الدقة الإيجابية 1.0 (Precision)، ومعدل الاستدعاء (Recall) 0.9982، مما أسفر عن قيمة F1 بلغت 0.9991، و ROC-AU قيمة 0.9598 مما يعكس كفاءة عالية كما في الشكل (5):

```
- **Dataset**: CICDDoS2019

## Performance Metrics
- **Accuracy**: 0.9981
- **Precision**: 1.0000
- **Recall**: 0.9982
- **F1 Score**: 0.9991
- **ROC AUC**: 0.9598
- **Optimal Threshold**: 0.3785

## Features
['flow_duration', 'flow_IAT_mean', 'flow_bytes', 'flow_packets']
```

الشكل (5) نتائج تدريب الخوارزمية

**4.3. اختبار النظام المقترح:**

تمت محاكاة النظام المقترح ضمن بيئة Mininet[12] لإنشاء طوبولوجيا شبكية واضحة تضم أربعة مبدلات (Switches) من نوع Open vSwitch (OVS) وثمانية مضيفين (Hosts)، حيث جرى ربطها بمتحكم Ryu. تم تخصيص المضيف h2 ليكون الضحية (IP = 10.0.0.2) ، بينما تم توزيع المضيفين الآخرين على مبدلات مختلفة لتوليد حركة مرور هجومية ومعتادة كما في الشكل (6).

يرتكز اختيار هذا التصميم على تحقيق هدفين أساسيين:

1. محاكاة واقعية لهجمات DDoS الموزعة:

تتميز هجمات حجب الخدمة الموزعة بطبيعتها المنتشرة جغرافياً وشبكياً، حيث تعتمد على عدد كبير من المصادر التي تولّد حركة مرور ضارة باتجاه الهدف. يمنح توزيع المضيفين المهاجمين على مبدّلات متعددة نموذجاً أقرب إلى السيناريو الواقعي للهجمات، بما يعزز مصداقية ودقة التقييم.

2. اختبار قدرة النظام على التعامل مع بيئة شبكية معقدة:

يتيح وجود عدة مبدّلات تقييم آلية الكشف ضمن سياق شبكي متعدد المسارات، حيث تمر حركة المرور عبر مسارات مختلفة نحو الهدف. يسهم ذلك في اختبار قدرة النظام على تحليل تدفقات البيانات في بيئة غير مبسطة، ما يوفر مؤشراً أقرب إلى أدائه في شبكات حقيقية ذات بنية معقدة وموزعة.

تم تصميم الهجوم الموزّع (DDoS) بحيث يصدر من عدة مضيفين موزعين على مبدلات مختلفة، وذلك لمحاكاة السيناريوهات الواقعية للهجمات. على سبيل المثال، تم استخدام hping3 لتوليد هجمات UDP Flood من مصادر متعددة نحو الضحية h2 كالتالي:

```
h1 hping3 --udp -p 80 --flood 10.0.0.2
```

```
h3 hping3 --udp -p 80 --flood 10.0.0.2
```

```
h4 hping3 --udp -p 80 --flood 10.0.0.2
```

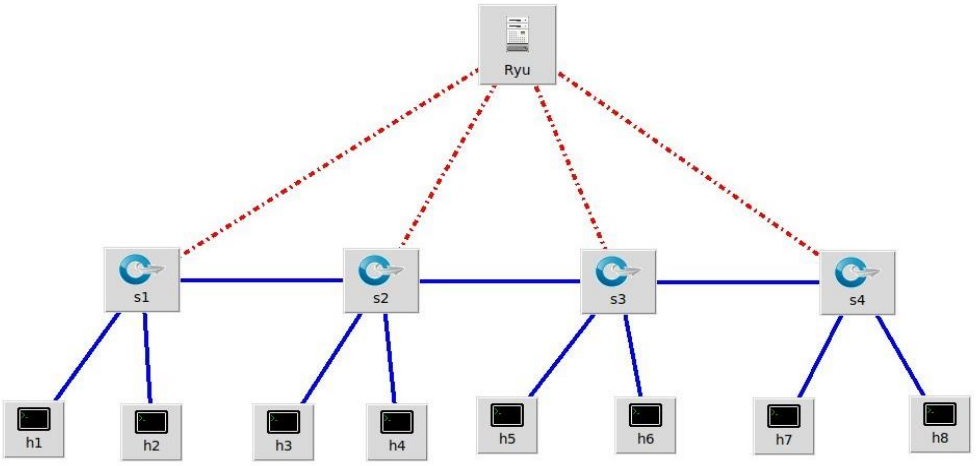
تم توليد هجمات Syn Flood من مصادر متعددة نحو الضحية h2 كالتالي:

```
h1 hping3 -p 80 --flood 10.0.0.2
```

h3 hping3 -p 80 --flood 10.0.0.2

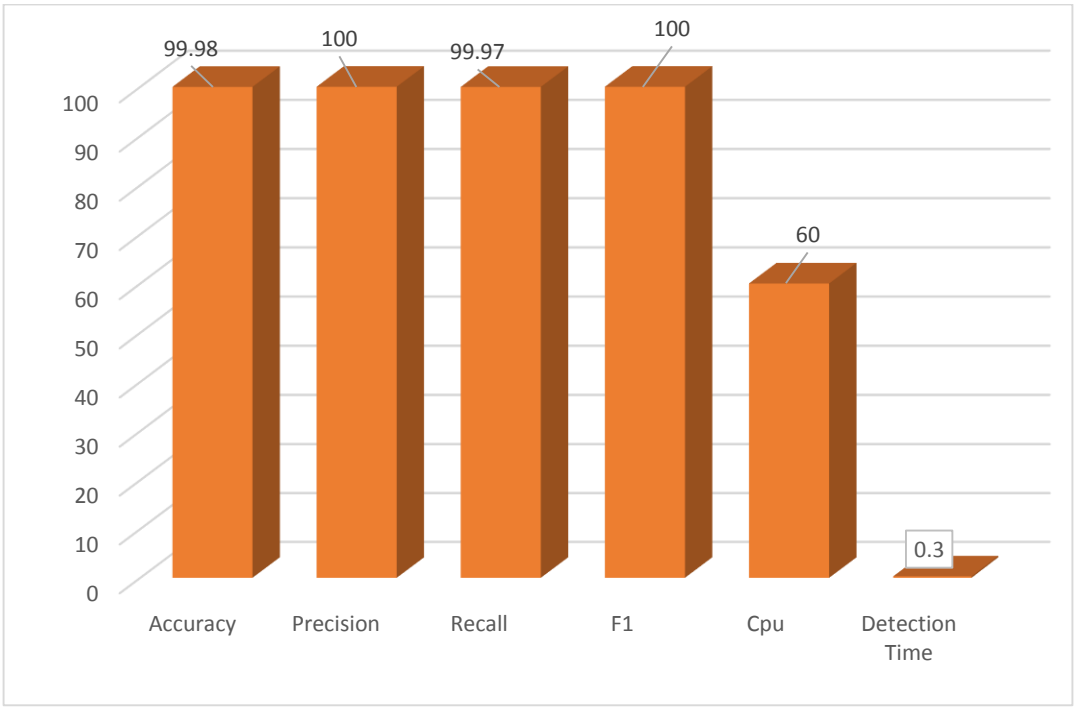
h4 hping3 -p 80 --flood 10.0.0.2

يضمن هذا توزيع الحمل الهجومي على عدة نقاط داخل الشبكة واختبار قدرة النظام على الكشف اللحظي والاستجابة السريعة .كما تم التحقق من أن عنوان IP للضحية 10.0.0.2 موجود فعلياً ضمن الطوبولوجيا، وتم مراقبة جميع المضيفين الهجوميين لضمان توليد حركة مرور شبكية متزامنة .




الشكل (6) طوبولجيا اختبارالنظام المقترح

تم تقييم النظام باستخدام مجموعة من المعايير المهمة منها الدقة (Accuracy)، ومعدل F1-score ، وزمن اكتشاف الهجوم، بالإضافة إلى استهلاك الذاكرة والمعالج كما في الشكل (7)، فضلاً عن القدرة على اكتشاف نوعي الهجوم المذكورين. أظهرت النتائج التي حققها النظام فعالية عالية وأداءً متميزاً في الكشف عن الهجمات ضمن بيئة شبكات SDN.



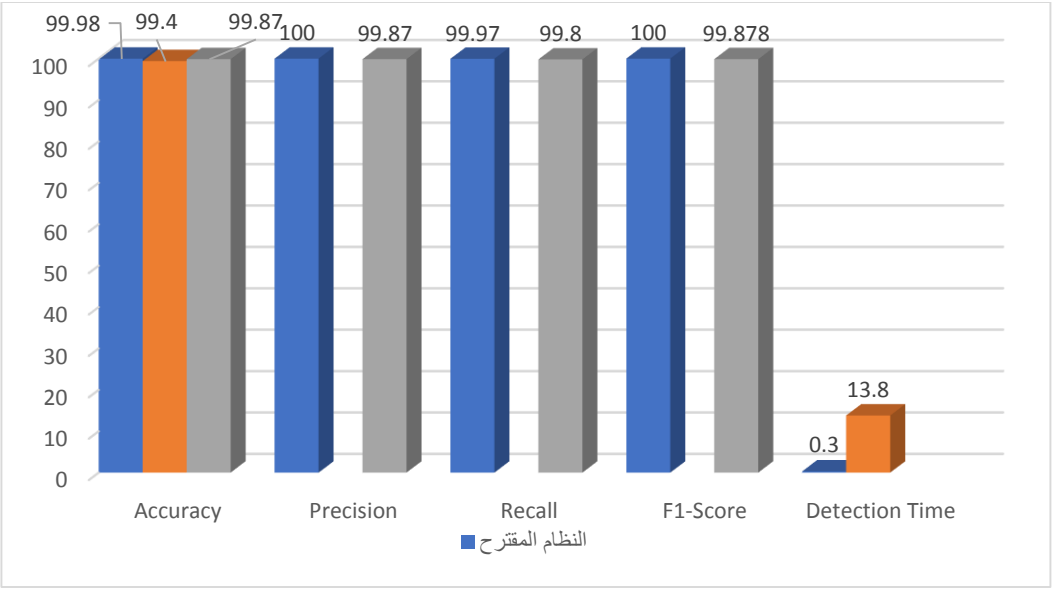

الشكل (7) معايير تقييم النظام المقترح

### 5.3. مقارنة مع الدراسات السابقة

ركزت الدراسة [4] على تصميم نظام معماري مرن وقابل للتوسع للكشف عن هجمات DDoS في طبقتي النقل والتطبيق، باستخدام مجموعتي البيانات CICDDoS2019 و CICDoS2017. أظهرت النتائج تفوق نماذج GRU و LSTM بدقة تجاوزت 99% ومعدل اكتشاف يفوق 95%، مع كفاءة عالية في استهلاك الموارد وسرعة المعالجة ضمن بيئة محاكاة واقعية.

أما الدراسة [6] فقد قارنت بين نماذج التعلم العميق LSTM، CNN والخوارزميات التقليدية KNN ، SVM، Naive Bayes ضمن بيئة محاكاة Mininet، حيث حقق نموذج KNN أعلى دقة (99.4%)، في حين أظهر LSTM أفضل توازن بين الدقة والاستدعاء، مما يجعله أكثر

موثوقية في الكشف شبه الحقيقي. وفي إطار تحسين الدقة والتعميم. يوضح الشكل (8) مقارنة بين الدراسات السابقة والنظام المقترح.




الشكل (8) مقارنة بين النظام المقترح مع الدراسات السابقة

يوضّح الجدول (2) تلخيص معايير المقارنة بين النظام المقترح والأعمال السابقة من حيث الدقة (Accuracy)، زمن الكشف و نسبة الدقة الإيجابية (Precision)، ومعدل الاستدعاء (Recall) ، وقيم F1، حيث أن بعض الأعمال لم تستخدم كامل المعايير في تقييم عملها مثل F1 وغيرها.

جدول (2) مقارنة بين النظام المقترح مع الدراسات السابقة

| | Accuracy | Precision | Recall | F1 | زمن الكشف |
|---|---|---|---|---|---|
| النظام المقترح | 99.98% | 100% | 99.97 | 100% | 0.3s |

| | | | | | |
|---|---|---|---|---|---|
| [4] | 99.877% | 99.879% | 99.877% | 99.878% | - |
| [6] | 99.4% | - | - | - | 13.8s |

### 6.3. تحليل النتائج

بعد إجراء الاختبارات والتقييمات للنظام المقترح في بيئة المحاكاة، تظهر النتائج التي تم الحصول عليها أداءً استثنائياً من حيث الدقة والسرعة.

#### 1. تحليل دقة وكفاءة النموذج (فعالية الكشف)

أظهر النموذج المقترح، الذي يجمع بين خوارزمية XGBoost وآلية العتبة التكيفية، قيماً متقدمة لمقاييس الأداء، حيث تجاوزت الدقة (Accuracy) ومعدل F1 والدقة النوعية (Precision) والاستدعاء (Recall) حاجز 99.9%. يمكن تفسير هذا التفوق من خلال العوامل التالية:

- قوة خوارزمية XGBoost: تمتاز خوارزمية XGBoost بقدرتها الفائقة على التعامل مع البيانات المنطقية والرقمية، والتعلم من العلاقات غير الخطية بين ميزات التدفق الشبكي. تمتاز أيضاً بقدرتها على تجنب الإفراط في التخصيص (Overfitting) من خلال التنظيم (Regularization)، إلى جانب آلية التعزيز (Boosting) التي تعمل على تصحيح أخطاء النماذج السابقة، جعلتها هذه الميزات قادرة على بناء نموذج معمم ودقيق للغاية للتمييز بين حركة المرور الطبيعية وهجمات DDoS.
- جودة وملاءمة الميزات المستخرجة: لم يعتمد النظام على تحليل الحزم بشكل فردي، بل على الميزات الإحصائية للتدفق مثل flow_IAT_mean، flow_packets، flow_bytes، و flow_duration. تعكس هذه الميزات بشكل مباشر السلوك الشاذ لهجمات DDoS.

- فعالية آلية العتبة التكيفية (Adaptive Thresholding): ساهمت هذه الآلية بشكل كبير في تحقيق معدل دقة إيجابية (Precision) بنسبة 100% بدلاً من استخدام عتبة ثابتة قد تؤدي إلى إنذارات كاذبة (False Positives) أثناء فترات الذروة الطبيعية للشبكة. سمحت الآلية الديناميكية بتعديل عتبة التصنيف بناءً على الأداء اللحظي. هذا يعني أن النظام أصبح أكثر تشدداً أو تساهلاً تلقائياً، مما يحافظ على دقة الكشف العالية ويقلل من اضطرابات الشبكة الناتجة عن الحجب الخاطئ للتدفقات المشروعة.

**2. تحليل سرعة الاستجابة (زمن الكشف)**

سجل النظام زمن كشف لم يتجاوز 0.3 ثانية لأغلب الهجمات المُحاكاة. تعزى هذه السرعة الفائقة، التي تعتبر حاسمة لمواجهة هجمات DDoS ، إلى العمارة المبتكرة للنظام:

- **التكامل بين P4 و وحدة التحكم:** تم تنفيذ الجزء الأكثر استهلاكاً للوقت، وهو جمع وإحصاء بيانات التدفق، داخل مبدل P4 نفسه على مستوى البيانات. هذا يلغي الحاجة إلى إرسال كل حزمة فردية إلى المتحكم لتحليلها، مما يقلل بشكل كبير من الحمل على قناة الاتصال Southbound وعنق الزجاجة في وحدة التحكم. عند وصول العينة الإحصائية (Sampling) إلى المتحكم، يكون نموذج XGBoost المدرب مسبقاً جاهزاً لاتخاذ القرار فوراً، حيث أن عملية التصنيف بالنسبة لنموذج مدرب هي عملية سريعة جداً.

- **مقارنة مع الدراسات السابقة:** عند مقارنة زمن الكشف (0.3 ثانية) مع أعمال سابقة مثل[6] التي سجلت 13.8 ثانية، يتضح الفرق الجوهري الذي يحدثه استخراج الميزات في مستوى البيانات . تعتمد غالبية الحلول التقليدية على تجميع البيانات في المتحكم أولاً ثم تحليلها، مما يؤدي إلى تأخير أكبر في الكشف.

**3. تحليل الأداء في مواجهة أنواع الهجمات المختلفة**

أثبت النظام فاعليته في الكشف عن كلاً من هجومي SYN Flood و UDP Flood. يعتمد آلية الكشف على السلوك الإحصائي للهجوم. نظراً لأن كلا النوعين ينتجان زيادة مفاجئة وكبيرة في عدد الحزم مع انخفاض في الفترات الزمنية بينها، فإن النموذج قادر على تصنيفها كحالة شاذة بغض النظر عن البروتوكول المستهدف TCP أو UDP. تجعل هذه الخاصية النظام مرناً وقابلاً للتكيف مع أنواع عديدة من هجمات الفيضان (Flood Attacks).

## 4. الخلاصة والآفاق المستقبلية:

قدّم هذا البحث إطاراً متكاملاً للكشف اللحظي عن هجمات حجب الخدمة الموزعة (DDoS) في شبكات SDN من خلال دمج برمجة مستوى البيانات بلغة P4 مع نموذج XGBoost وآلية العتبة التكيفية. أظهرت النتائج دقة عالية تجاوزت 99.9% وزمن كشف لا يتعدى 0.3 ثانية، مما يؤكد فعالية النظام في البيئات التي تتطلب استجابة فورية. رغم ذلك، يبقى الأداء مقيداً ببيئة المحاكاة ونطاق الهجمات المختبرة، إضافةً إلى التحديات المرتبطة باستهلاك موارد مبدلات P4 واعتماد النظام على متحكم مركزي واحد. تفتح هذه الدراسة المجال لأعمال مستقبلية تشمل توسيع الاختبارات لتشمل هجمات أكثر تنوعاً، وتطبيق النظام في بيئات متعددة المتحكمات، إضافةً إلى دمج تقنيات التعلم المستمر والتعلم بالنقل لتعزيز قدرة النموذج على التكيف مع التغيرات المستمرة في أنماط الهجمات.

## 5. جدول المختصرات

| | |
|---|---|
| SDN | Software-Defined Networking |
| DDoS | Distributed Denial-of-Service |
| P4 | Programming Protocol-Independent Packet Processors |
| XGBoost | Extreme Gradient Boosting |
| API | Application Programming Interface |
| ROC-AUC | Receiver Operating Characteristic - Area Under Curve |

| | |
|---|---|
| SMOTE | Synthetic Minority Over-sampling Technique |

## 6. المراجع العلمية:

[1] الضاهر، م. (2021). تحليل أداء وحدات تحكم الشبكات المعرفة برمجياً POX : و OpenDaylight. *مجلة جامعة البعث*، (17)*43*، 101-118.

[2] lubaidan, H., Alzaher, R., AlQhatani, M., & Mohammed, R. (2023). DDoS detection in software-defined network (SDN) using machine learning. In Dhinaharan Nagamalai (Ed.), *International Journal on Cybernetics & Informatics (IJCI)* (Vol. 12, No. 4, pp. 93–104).

[3] Chuang, H.-M., Liu, F., & Tsai, C.-H. (2022). Early detection of abnormal attacks in software-defined networking using machine learning approaches. *Symmetry, 14*(6), 1178.

[4] Yungaicela, N. M., Vargas-Rosales, C., & Pérez, J. A. (2021). *SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning*. *IEEE Access, 9*.

[5] Gebremeskel, T. G., Gemeda, K. A., Krishna, T. G., & Ramulu, P. J. (2023). *DDoS attack detection and classification using hybrid model for multicontroller SDN*. *Security and Communication Networks, 2023*, Article 9965945.

[6] adze, J. D., Bamfo-Asante, A. A., Agyemang, J. O., Nunoo-Mensah, H., & Opare, K. A.-B. (2021). *An investigation into the application of deep learning in the detection and mitigation of DDoS attack on SDN controllers*. *Technologies, 9*(1), 14.

[7] Wang, H., & Li, W. (2021). *DDosTC: A transformer-based network attack detection hybrid mechanism in SDN*. *Sensors, 21*(15), 5047.

[8] Wang, J., Wang, L., & Wang, R. (2023). *A method of DDoS attack detection and mitigation for the comprehensive coordinated protection of SDN controllers*. *Entropy, 25*(8), 1210.

[9] Carvalho, R. N., et al. (2021). Detecting DDoS attacks on SDN data plane with machine learning. In Ninth International Symposium on Computing and Networking Workshops (CANDARW).

[10] Ali, Z. A., Abduljabbar, Z. H., Tahir, H. A., Sallow, A. B., & Almufti, S. M. (2023). Exploring the power of eXtreme Gradient Boosting algorithm in machine learning: A review. *Academic Journal of Nawroz University, 12*(2), 320–?.

[11] Sharafaldin, I., et al. (2019). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. IEEE.

[12] Mininet Team. (2022). *Mininet: An Instant Virtual Network on your Laptop (or other PC)*. Retrieved from http://mininet.org.

[13] Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., & Walker, D. (2014).

P4: Programming protocol-independent packet processors. *ACM SIGCOMM Computer Communication Review, 44*(3), 87–95.

[14] Sheikh, M. N. A., Hwang, I.-S., Raza, M. S., & Ab-Rahman, M. S. (2024). *A qualitative and comparative performance assessment of logically centralized SDN controllers via Mininet emulator*. Computers, 13(4),85.

[15] Fan, Z., & You, Z. (2024). *Research on network intrusion detection based on XGBoost algorithm and multiple machine learning algorithms*. *Theoretical and Natural Science, 31.*