الجمهورية العربية السورية

جامعة دمشق

كلية الهندسة الميكانيكية والكهربائية

قسم هندسة الحواسيب والأتمتة

تطوير خوارزميات الذكاء الصنعي لاختيار بروتوكول التوجيه المناسب اعتماداً على العوامل المؤثرة

أطروحة مقدمة لنيل درجة الدكتوراه في هندسة الحواسيب وشبكاتها

إعداد

م. محمد عرفان الخطيب

إشراف

د.م. بسام محمد

2025-2024

لجنة الحكم

لجنة الحكم المشكلة بناءً على قرار مجلس البحث العلمي والدراسات العليا في جامعة دمشق رقم /237/ م.ب.د المتخذ بالجلسة رقم /2/ التي انعقدت بتاريخ 2025/10/6 لمناقشة رسالة الدكتوراه في كلية الهندسة الميكانيكية والكهربائية قسم هندسة الحواسيب والأتمتة التي أعدها الطالب محمد عرفان الخطيب بعنوان:

" تطوير خوارزميات الذكاء الصنعي لاختيار بروتوكول التوجيه المناسب اعتماداً على العوامل المؤثرة "

أ.د. سمير كرمان ----------

أستاذ في قسم هندسة الحواسيب والأتمتة عضواً

الاختصاص: نظرية المعلومات في الحاسب

د. بسام محمد ----------

أستاذ مساعد في قسم هندسة الحواسيب والأتمتة عضواً مشرفاً

الاختصاص: بنية الحاسبات الإلكترونية

د. محمد مازن المحايري ----------

أستاذ مساعد في قسم هندسة الحواسيب والأتمتة عضواً

الاختصاص: عناصر وبنية تنظيم الحاسبات

د. محمد الجنيدي ----------

باحث رئيسي في المعهد العالي للعلوم التطبيقية والتكنولوجيا عضواً

الاختصاص: أمن الشبكات الحاسوبية

د. هديل الغفري ----------

مدرّس في قسم هندسة الحواسيب والأتمتة عضواً

الاختصاص: شبكات لاسلكية وبروتوكولاتها

تصريح خطي

أنا الموقع أدناه، أصرح بعلمي الكامل وقبولي:

- أن كل ما ينتج عن البحث والأطروحة هو ملكية فكرية ومالية بالتشارك مع جامعة دمشق، وأنني ألتزم بأخذ موافقة الجامعة في حال رغبتي في نشر البحث أو الأطروحة أو جزء منها نصاً أو مضموناً خارج إطار الجامعة (من دور نشر أو مكتبات أو مواقع إلكترونية وغيرها من وسائل النشر).
- أنه لا يوجد أي جزء من هذه الأطروحة اقتُبس من عمل علمي آخر أنجز للحصول على شهادة أخرى في جامعة دمشق أو أية جامعة أو معهد تعليمي داخل أو خارج الجمهورية العربية السورية.

الاسم والتوقيع

الإهداء

- إلى أبي العزيز، الذي علّمني معنى الصبر، وغرس في نفسي حب العلم والعمل، فكان دائماً قدوتي ومصدر قوتي.
- إلى أمي الغالية، التي كان دعاؤها سرّ نجاحي، وابتسامتها ملاذي في كل لحظة صعبة.
- إلى زوجتي الحبيبة، شريكة حياتي، وسند أيامي، ورفيقة دربي في هذه الرحلة الطويلة. لقد كنتِ نورا أضاء طريقي، والدعم الذي منحني القوة للاستمرار، وهذا العمل هو ثمرة حبك وصبرك وتضحياتك التي لا تقدر بثمن.
- إلى أولادي الأحباء، مصدر البهجة والأمل، الذين منحوني الدافع لأكون أفضل، وأن أواصل الطريق مهما كان شاقاً.
- إلى إخوتي الأعزاء، الذين أحاطوني بالمحبة والدعم، وتذوقت معهم أجمل اللحظات.
- إلى عائلة زوجتي الكريمة، الذين غمروني بالمحبة وصدق المشاعر وشاركوني فرحتي بهذا الإنجاز.
- إلى أصدقائي، الذين أحمل لهم المودة في قلبي.
- إلى أساتذتي، الشموع التي تحترق لتضيء الآخرين.

إليهم جميعاً أهدي هذا العمل، عرفاناً وامتناناً، وتقديراً لا ينتهي.

شكر وتقدير

أتوجه بجزيل الشكر والامتنان إلى المشرف على هذا البحث

الدكتور المهندس بسام محمد

الذي لم يتوانى عن تقديم المساعدة العلمية لإنجاز هذا البحث.

كما وأتقدم بالشكر لأعضاء الهيئة التدريسية في قسم الحواسيب والأتمتة على دعمهم لي في مسيرتي العلمية وخاصة رئيس القسم **أ.د. سمير كرمان** الذي لم يدخر جهداً في تقديم العون لإتمام هذا البحث

الفهرس Index

قائمة الأشكال List of Figures.....10

قائمة الجداول List of Tables.....13

جدول المصطلحات Table of glossary.....14

ملخص الدراسة.....15

الفصل الأول مدخل إلى البحث.....16

1.1. مقدمة 17

2.1. مشكلة البحث 17

3.1. الغاية من البحث 17

4.1. أهمية البحث 18

5.1. منهجية البحث 18

الفصل الثاني الدراسة المرجعية 19

الفصل الثالث أساسيات خوارزمية تحسين مستعمرة النمل.....33

1.3. مقدمة عن خوارزمية تحسين مستعمرة النمل (ACO) 34

2.3. خوارزمية تحسين مستعمرة النمل 34

3.3. تحديث الفيرمون.....36

4.3. خواص وسلوك الخوارزمية ACO 37

1.4.3. تغذية راجعة إيجابية و تبخر 37

2.4.3. تأثير المعاملات α,β والركود. 37

3.4.3. عدد النمل وقوائم الجوار المرشحة.....37

4.4.3. الأساس البيولوجي (تجارب الجسرين).....38

5.3. خطوات خوارزمية (ACO) الأساسية 39

الفصل الرابع البحث بالعمق أولاً 40

1.4. مقدمة لخوارزمية البحث بالعمق أولاً 41

2.4. آلية العمل 41

الفصل الخامس بروتوكولات التوجيه في الشبكات الحاسوبية.....46

1.5. مقدمة لبروتوكولات التوجيه بالشبكات الحاسوبية..... 47
2.5. أهمية بروتوكولات التوجيه 47
3.5. التوجيه Routing..... 48
4.5. أنواع بروتوكولات التوجيه 49
5.5. خوارزميات التوجيه..... 51
5. 5. 1. خوارزمية شعاع المسافة Distance Vector..... 51
1.1.5.5. اكتشاف الطريق والاختيار 52
2.1.5.5. بروتوكولات شعاع المسافة 55
5. 5. 2. التوجيه حسب حالة الوصلة 58
5. 5. 3. خوارزمية ديكسترا (Dijkstra) 59
4.5.5. بروتوكولات حالة الوصلة..... 64
5.5.5. فوائد بروتوكولات حالة الوصلة..... 65
6.5.5. البروتوكول Open Shortest Path First (OSPF) 65
1.6.5.5. مصطلحات البرتوكولOSPF..... 65
2.6.5.5. حساب شجرة المسار الأقصر أولاً (SPF tree)..... 67
7.5.5. خوارزمية شعاع المسافة المحسّن..... 67
5. 5. 7. 1. بروتوكولات شعاع المسافة المحسّن..... 68
8.5.5. الفرق بين بروتوكولات حالة الوصلة وشعاع المسافة..... 71
الفصل السادس دراسة تأثير بروتوكولات التوجيه عند نقل أنواع البيانات المختلفة..... 73
1.6. مقدمة لأنواع البيانات..... 74
2.6. العوامل المؤثرة في أداء بروتوكولات التوجيه في هذه الدراسة 75
3.6. دراسة تأثير بروتوكول التوجيه عند نقل البيانات النصية..... 77
1.3.6. محددات القياس المستخدمة في الدراسة..... 77
2.3.6. تحليل الشبكات التي تم دراستها..... 81
3.3.6. مواصفات البنية التحتية للشبكات..... 81
4.3.6. تقييم نتائج التحليل 84

4.6. دراسة تأثير بروتوكول التوجيه عند نقل الفيديو 89
1.4.6. محددات القياس المستخدمة في الدراسة 89
2.4.6. الشبكات المدروسة ضمن برنامج المحاكاة 90
3.4.6. جمع النتائج 91
4.4.6. تقييم وتحليل نتائج الدراسة 92
1.4.4.6. تحليل الشبكات بدلالة حجم الأطر 93
2.4.4.6. تحليل تباين تأخير حزم الفيديو في الشبكات 94
3.4.4.6. تحليل نسبة الضياع في الشبكات 95
4.4.4.6. تحليل زمن التأخير من طرف إلى طرف في الشبكات 96
5.4.4.6. تحليل أداء الشبكات مع تغير الزمن 96
5.4.6. الخلاصة 103
5.6. دراسة تأثير بروتوكول التوجيه عند نقل الصوت 104
1.5.6. محددات القياس ضمن الدراسة 104
2.5.6. نتائج التحليل 106
3.5.6. الاستنتاجات 108
الفصل السابع إنشاء وتطوير الخوارزمية المقترحة 110
1.7. مقدمة عن الخوارزمية المقترحة 111
2.7. آلية عمل الخوارزمية المقترحة 111
3.7. الخوارزمية المقترحة 112
4.7. حقول النملة المقترحة 115
5.7. أثر زيادة عدد النمل على اكتشاف المسارات 117
6.7. نتائج المحاكاة 118
7.7. الاستنتاجات 121
8.7. التطوير الأول للخوارزمية المقترحة 122
1.8.7. إضافة الآلية DFS (Depth-First-Search) 122
1.1.8.7. ميزات الدمج 123

2.1.8.7. الفرق بين DFS التقليدية والمطبقة داخل ACO المطوّرة..... 124

2.8.7. إضافة تقييم جودة الوصلات 124

1.2.8.7. آلية تحديث الفيرمونات وتكيّفها مع الجودة 124

2.2.8.7. حساب جودة الخدمة (Q) 125

3.2.8.7. الصيغة المعدلة مع معاملات (k_i) 126

4.2.8.7. تحديث الفيرمونات 128

5.2.8.7. قياس الفيرمون..... 128

6.2.8.7. الأوزان والتطبيع..... 132

3.8.7. الخوارزمية المقترحة مع التطوير الأول..... 133

4.8.7. نتائج المحاكاة 134

5.8.7. الاستنتاجات 137

6.8.7. المقارنة بين الخوارزمية ACO المطورة والكلاسيكية..... 138

7.8.7. مقارنة نتائج الخوارزمية المطورة مع الكلاسيكية 139

8.8.7. تحليل النتائج..... 141

9.8.7. مقارنة مرجعية للمرجع (Shahid et al. 2024) [8] مع الخوارزمية المقترحة..... 141

9.7. الخوارزمية المقترحة النهائية مع التطوير الثاني..... 145

1.9.7. مخطط الخوارزمية المقترحة النهائية..... 148

الفصل الثامن الإسهامات العلمية والأفق المستقبلية..... 149

1.8. خلاصة البحث..... 150

2.8. الإسهامات العلمية للبحث..... 151

3.8. التطبيقات المقترحة للبحث 152

4.8. التوصيات والآفاق المستقبلية..... 154

المراجع Refrences..... 155

قائمة الأشكال List of Figures

الشكل (3–1): تجربة الجسرين 38
الشكل (3–2): مخطط خوارزمية ACO 39
الشكل (4–1): البحث بالعمق في خوارزمية DFS.....44
الشكل (5–1): توجيه البروتوكولات 49
الشكل (5–2): الفرق بين التوجيه الداخلي والخارجي.....50
الشكل (5–3): خوارزمية شعاع المسافة 52
الشكل (5–4): تنفيذ خوارزمية بيلمان–فورد.....54
الشكل (5–5): التوجيه حسب حالة الوصلة.....59
الشكل (5–6): تنفيذ خوارزمية ديكسترا.....61
الشكل (5–7): البنية الهرمية لبروتوكولات حالة الوصلة 63
الشكل (6–1): مكونات الشبكة المركزية.....81
الشكل (6–2): شبكة نجمية.....83
الشكل (6–3): شبكة حلقية.....83
الشكل (6–4): شبكة هجينة.....83
الشكل (6–5): شبكة هرمية.....83
الشكل (6–6): شبكة متشابكة (Mesh) كبيرة.....83
الشكل (6–7): شبكة هجينة كبيرة.....83
الشكل (6–8): شبكة متشابكة (Mesh) صغيرة.....84
الشكل (6–9): شبكة خطية.....84
الشكل (6–10): نسبة عدد المرات التي حققت فيها بروتوكولات التوجيه أفضل أداء عند نقل البيانات النصية.....86
الشكل (6–11): عدد المرات الكلي التي حقق فيها بروتوكول التوجيه أفضل أداء عند نقل البيانات النصية.....87
الشكل (6–12): زمن التأخير من طرف لطرف عند نقل الفيديو (سيناريو RIP).....99
الشكل (6–13): تباين التأخير عند نقل الفيديو (سيناريو RIP).....99
الشكل (6–14): الإنتاجية عند نقل الفيديو (سيناريو RIP).....99
الشكل (6–15): زمن التأخير من طرف لطرف عند نقل الفيديو (سيناريو OSPF).....100
الشكل (6–16): تباين التأخير عند نقل الفيديو (سيناريو OSPF).....100

الشكل (6–17): الإنتاجية عند نقل الفيديو (سيناريو OSPF).....100
الشكل (6–18): زمن التأخير من طرف لطرف عند نقل الفيديو (سيناريو EIGRP).....100
الشكل (6–19): تباين التأخير عند نقل الفيديو (سيناريو EIGRP).....100
الشكل (6–20): الإنتاجية عند نقل الفيديو (سيناريو EIGRP).....101
الشكل (6–21): نسبة عدد الشبكات التي حققت فيها بروتوكولات التوجيه أفضل أداء عند نقل الفيديو.....102
الشكل (6–22): عدد المرات الكلي التي حقق فيها بروتوكول التوجيه أفضل أداء عند نقل الفيديو.....102
الشكل (6–23): نسبة عدد الشبكات التي حققت فيها بروتوكولات التوجيه أفضل أداء عند نقل الصوت.....107
الشكل (6–24): عدد المرات الكلي التي حقق فيها بروتوكول التوجيه أفضل أداء عند نقل الصوت.....108
الشكل (7–1): مراحل دورة حياة النمل بالخوارزمية المقترحة 114
الشكل (7–2): حقول النملة بالخوارزمية المقترحة 115
الشكل (7–3): مخطط الخوارزمية المقترحة 116
الشكل (7–4): علاقة زيادة عدد النمل على اكتشاف المسارات.....118
الشكل (7–5): جزء من محاكاة المرحلة الأولى.....118
الشكل (7–6): اختيار المسارات في المرحلة الأولى.....119
الشكل (7–7): اختيار المسارات في المرحلة الثانية.....119
الشكل (7–8): قياس الإنتاجية للخوارزمية المقترحة 120
الشكل (7–9): قياس التأخير للخوارزمية المقترحة.....120
الشكل (7–10): قياس زمن الاستجابة للخوارزمية المقترحة.....121
الشكل (7–11): قياس التذبذب للخوارزمية المقترحة.....121
الشكل (7–13): مخطط حساب الجودة في الخوارزمية المقترحة.....125
الشكل (7–14): علاقة الفيرمون مع محددات مثل الضياع والتذبذب ضمن الشبكة.....131
الشكل (7–15): علاقة k_i بالجودة وتحديد منطقة العمل.....132
الشكل (7–16): مخطط الخوارزمية المقترحة بعد التطوير الأول.....134
الشكل (7–17): جزء من محاكاة المرحلة الأولى.....134
الشكل (7–18): اختيار المسارات في المرحلة الأولى.....135
الشكل (7–19): اختيار المسارات في المرحلة الثانية.....135
الشكل (7–20): قياس الإنتاجية بعد التطوير الأول.....137
الشكل (7–21): قياس التأخير بعد التطوير الأول.....137

الشكل (7–22): قياس زمن الاستجابة بعد التطوير الأول.....137
الشكل (7–23): قياس التذبذب بعد التطوير الأول.....137
الشكل (7–24): قياس التذبذب عند المقارنة مع الخوارزمية الكلاسيكية.....140
الشكل (7–25): قياس التأخير عند المقارنة مع الخوارزمية الكلاسيكية.....140
الشكل (7–26): قياس زمن الاستجابة عند المقارنة مع الخوارزمية الكلاسيكية.....140
الشكل (7–27): قياس الإنتاجية عند المقارنة مع الخوارزمية الكلاسيكية.....140
الشكل (7–28): قياس نسبة التوصيل عند المقارنة مع الخوارزمية الكلاسيكية.....140
الشكل (7–29): اختيار المسارات عند فشل وصلة.....142
الشكل (7–30): اختيار المسارات في الشبكة المرجعية بالحالة العادية.....143
الشكل (7–31): اختيار المسارات في الشبكة المرجعية عند فشل وصلة.....143
الشكل (7–32): مخطط الخوارزمية المقترحة بعد التطوير الثاني.....148

قائمة الجداول List of Tables

الجدول (5–1): مقارنة بين بروتوكولات شعاع المسافة وحالة الوصلة 72
الجدول (6–1): نسبة عدد المرات التي كان فيها كل بروتوكول أفضل أداءً عند نقل البيانات النصية.....86
الجدول (6–2): علاقة حجم الأطر المرسلة مع أداء بروتوكولات التوجيه حسب الطوبولوجيا عند نقل الفيديو.....94
الجدول (6–3): قيم المحددات للبروتوكول RIP مع تغير زمن المحاكاة عند نقل الفيديو.....97
الجدول (6–4): قيم المحددات للبروتوكول OSPF مع تغير زمن المحاكاة عند نقل الفيديو.....97
الجدول (6–5): قيم المحددات للبروتوكول EIGRP مع تغير زمن المحاكاة عند نقل الفيديو.....97
الجدول (6–6): نسبة عدد المرات التي تفوّق بها كل بروتوكول حسب محددات الدراسة عند نقل الفيديو.....101
الجدول (6–7): نسبة عدد المرات التي تفوّق فيها كل بروتوكول توجيه حسب محددات الدراسة عند نقل الصوت.107
الجدول (7–1): تفصيل حقول النملة بالخوارزمية المقترحة.....115
الجدول (7–2): نتائج مقارنة الخوارزمية المقترحة مع نتائج بروتوكولات التوجيه.....120
الجدول (7–3): مقارنة بين خوارزمية DFS التقليدية وبين المطبقة داخل ACO المطورة.....124
الجدول (7–4): أوزان الخدمات حسب ITU.....132
الجدول (7–5): مقارنة الخوارزمية المقترحة المطورة مع نتائج بروتوكولات التوجيه.....136
الجدول (7–6): مقارنة بين ACO الكلاسيكية و ACO المقترحة بعد التطوير الأول.....138
الجدول (7–7): مقارنة بين نتائج ACO الكلاسيكية وACO المقترحة بعد التطوير الأول.....140
الجدول (7–8): مقارنة بين أداء الخوارزمية المقترحة وبين بروتوكولات التوجيه في الشبكة المرجعية.....144

جدول المصطلحات Glossary

المصطلح	المعنى باللغة الانكليزية	المعنى باللغة العربية
ACK	Acknowledgement	إشعار الاستلام
ACO	Ant Colony Optimization	تحسين مستعمرة النمل
AS	Autonomous system	النظام المستقل
BDR	Backup designated router	الموجّه المعيَّن الاحتياطي
BGP	Border Gateway Protocol	بروتوكول البوابة الحدودية
BW	Band Width	عرض النطاق الترددي
DB	Data Base	قاعدة البيانات
DFS	Depth-First-Search	البحث بالعمق أولاً
DR	Designated router	الموجّه المعيَّن
DUAL	Diffusing Update Algorithm	خوارزمية التحديث الانتشاري
EGP	Exterior Gateway Protocol	بروتوكول البوابة الخارجية
EIGRP	Enhanced Interior Gateway Routing Protocol	بروتوكول البوابة الداخلية المحسَّن
FTP	File Transfer Protocol	بروتوكول نقل الملفات
IGP	Interior gateway protocol	بروتوكول البوابة الداخلية
IP	Internet Protocol	بروتوكول الانترنت
ITU	International Telecommunication Union	الاتحاد الدولي للاتصالات
LSA	Link State Advertisement	إعلان حالة الوصلة
MOS	Mean Opinion Score	متوسط تقييم الرأي
OSPF	Open Shortest Path First	أقصر مسار مفتوح أولاً
QoS	Quality of service	جودة الخدمة
RIP	Routing Information Protocol	بروتوكول معلومات التوجيه
RTT	Round Trip Time	زمن الرحلة ذهاباً وإياباً
SDN	Software-Defined Networking	الشبكات المعرفة بالبرمجيات
SPF	Shortest Path First	أقصر مسار أولاً
TCP	Transmission Control Protocol	بروتوكول التحكم في الإرسال
TSP	Travelling salesman problem	مشكلة البائع المتجول
UDP	User Datagram Protocol	بروتوكول مخطط بيانات المستخدم
UTP	Unshielded Twisted Pair	زوج مجدول غير محمي
VoIP	Voice over Internet Protocol	الصوت عبر بروتوكول الإنترنت
WAN	Wide Area Network	شبكة واسعة النطاق

ملخص الدراسة

تشكل شبكات الاتصالات الحاسوبية العمود الفقري للبنية التحتية الأساسية لنقل وتبادل المعلومات في يومنا هذا، حيث تتنوع الخدمات التي تقدمها بين الاتصال الصوتي والفيديو ونقل الملفات والتطبيقات التفاعلية وغيرها. ولضمان نقل وتبادل البيانات بكفاءة وموثوقية، تعتمد هذه الشبكات على العديد من بروتوكولات التوجيه مثل OSPF (Open Shortest Path First) و EIGRP (Enhanced Interior Gateway Routing Protocol) و RIP (Routing Information Protocol)، والتي تُحدّد المسارات التي تسلكها حزم البيانات بين المصدر والهدف. وبالرغم من فعالية هذه البروتوكولات، إلا أنها عادةً ما تُطبّق على كامل الشبكة دون تمييز بين أنواع البيانات المختلفة أو متطلبات كل خدمة، مما قد يؤثر على أداء الشبكة لاحقاً.

في هذا البحث، اقترحنا خوارزمية هجينة ومطوّرة مستوحاة من خوارزمية تحسين مستعمرة النمل Ant Colony Optimization (ACO) ، تهدف إلى جعل عملية التوجيه أكثر ذكاءً وتكيّفاً مع خصائص البيانات، حيث تقوم باكتشاف جميع المسارات الكاملة الممكنة بين المصدر والهدف، ومن ثم تقوم بتطبيق بروتوكولات التوجيه المناسبة عليها: OSPF أو EIGRP، بناءً على طبيعة البيانات وخصائص الخدمة المطلوبة.

إن الفكرة الرئيسة للخوارزمية المطوّرة تقوم على دمج قوة بروتوكولات التوجيه التقليدية مع مرونة خوارزميات الذكاء الصنعي، بما يتيح تحسين كفاءة الخدمات التي تقدمها الشبكات بشكل عام وتقديم أداء أكثر توافقاً مع متطلبات التطبيقات المتنوعة.

الكلمات المفتاحية: خوارزميات الذكاء الصنعي، بروتوكولات التوجيه الداخلية OSPF، EIGRP، RIP، Ant Colony Optimization (ACO)

الفصل الأول

مدخل إلى البحث

Introduction to the Study

مقدمة:

أصبحت شبكات الاتصال الحاسوبية في عصرنا الحالي الركيزة الأساسية لمعظم الأنشطة التقنية والاقتصادية والاجتماعية حول العالم، فهي تسمح بتقديم وتبادل المعلومات والخدمات المتعلقة بها بشكل لحظي تقريباً مع توفير البنية التحتية التي تعتمد عليها المؤسسات والأفراد في مجالات متعددة مثل الاتصالات، التعليم عن بعد، الرعاية الصحية، والتجارة الإلكترونية. ومع التزايد المستمر في الحاجة إلى نقل بيانات متنوعة وبأحجام كبيرة، باتت الشبكات تواجه تحديات متزايدة في مجال ضمان جودة الخدمة QoS (Quality of service) مع الاستجابة لتلبية متطلبات كل نوع من أنواع البيانات المنقولة.

انطلاقاً من هذه الحاجة، يهدف هذا البحث إلى تطوير خوارزمية مستوحاة من الذكاء الجماعي لمستعمرات النمل، تهدف إلى دمج بروتوكولات التوجيه التقليدية مع آلية ذكية لتخصيص الشبكات والمسارات ضمنها وفقاً لنوع البيانات ومتطلباتها. وتأتي أهمية هذه الخوارزمية المطورة من كونها لا تستبدل البروتوكولات القائمة، بل تعيد تنظيم فضاء المسارات وتوزيعه بما يتناسب مع خصائص الخدمة المطلوب تنفيذها، مما يفتح آفاقاً لتحسين الأداء الكلي للشبكة وزيادة مرونتها في التعامل مع أنماط حركة مرور البيانات المختلفة التي يجب نقلها وتبادلها بين المصدر والهدف.

مشكلة البحث:

الحاجة إلى تطوير آلية ذكية قادرة على تحديد بروتوكول التوجيه المناسب لشبكة ما أو لمسارات محددة ضمنها استناداً إلى نوع البيانات المنقولة، بحيث يتم الاستفادة من نقاط القوة في كل بروتوكول وتوظيفها بما يتوافق مع خصائص الخدمة المطلوب تنفيذها. إن غياب مثل هذه الآلية يعني استمرار الاعتماد على أسلوب تقليدي غير تكيفي قد يحدّ من كفاءة الشبكة ويُضعف جودة الخدمات المقدمة عبرها.

الغاية من البحث :

الوصول إلى خوارزمية تحدد بروتوكول التوجيه المناسب لشبكة ما بشكل آلي حسب العوامل التي تؤثر عليها الأمر الذي يؤدي إلى الاستغناء عن استخدام الخبراء البشريين في اختيار البروتوكول المناسب لهذه الشبكة أو تلك.

أهمية البحث:

تكمن أهمية هذا البحث من سعيه إلى تحسين وتطوير أداء شبكات الاتصالات الحاسوبية، حيث يعتبر واحداً من أهم وأبرز التحديات التي يواجهها مسؤولو الشبكات في ظل التزايد المستمر لحجم البيانات المطلوب نقلها وتبادلها بالإضافة إلى تزايد تعقيد البنى الشبكية. فاختيار بروتوكول التوجيه الأمثل أصبح ضرورة لضمان نقل البيانات بكفاءة وسلاسة، مع تحقيق الاستخدام الأمثل للموارد، مما ينعكس إيجاباً على كفاءة الشبكة وجودة الخدمات التي تقدمها ويضمن استجابة سريعة للمستخدمين. وعلى خلاف معظم الدراسات السابقة التي اقتصرت على تحليل شبكة واحدة أو شبكتين فقط، مما جعل نتائجها محدودة وقابلة للتقييد، فقد توسعنا في هذا البحث لتتم دراسة وتحليل 450 شبكة بأحجام وطوبولوجيا مختلفة، الأمر الذي أتاح الوصول إلى نتائج أكثر دقة وموضوعية. إضافةً إلى ذلك، يُعد هذا العمل رائداً من حيث دمج خوارزمية ACO مع بروتوكولات التوجيه التقليدية بهدف إدارة الشبكة بذكاء وتحسين أدائها بشكل ديناميكي، وهو جانب لم تتم معالجته بعد في الأبحاث السابقة.

منهجية البحث:

اعتمد هذا البحث على منهجية تحليلية – تجريبية تقوم على تصميم ودراسة عدد كبير من الشبكات الحاسوبية بهدف الوصول إلى نتائج دقيقة وموثوقة حول البروتوكول الأنسب لكل نوع من أنواع البيانات ومن ثم تصميم وتطوير الخوارزمية المقترحة وفق خطوات منهجية علمية منظمة.

الفصل الثاني

الدراسة المرجعية

Literature Review

يعد اختيار أفضل مسار في الشبكة من أجل إرسال الحزم من المصدر إلى الهدف هي المهمة الأساسية للتأكد من جودة الاتصال وجودة الخدمة المطلوب تنفيذها، وللقيام بذلك، يتم إجراء عملية التوجيه على أجهزة التوجيه الشبكية لتحديد المسار المطلوب، وبما أنه توجد عدة بروتوكولات توجيه (ولكل واحد منهم آلية معينة باختيار المسار الأمثل) فإننا نجد العديد من الأبحاث التي قامت بدراسة اختيار بروتوكول التوجيه المناسب في الشبكة المدروسة من أجل توجيه الحزم بأفضل مسار وبأقل زمن تأخير وبأقل كلفة، ولكن هذه الأبحاث السابقة قامت بدراسة شبكة أو شبكتين وفق محددات دراسة مختلفة ربما تكون قليلة في بعض الأحيان على عكس ما سنقوم به في دراستنا هذه. ولتوضيح الفكرة بشكل أفضل يمكن أن نذكر على سبيل المثال لا الحصر ما قام به الباحثون (Monita et a., 2020) [1] في هذه الدراسة حيث قدموا دراسة معمّقة حول كيفية عمل الشبكات المعرفة بالبرمجيات (Software-Defined Networking) SDN عند تشغيل خدمات الوسائط المتعددة مثل المكالمات الصوتية ومكالمات الفيديو، مع التركيز على مقارنة أداء بروتوكولات التوجيه المختلفة. الفكرة الجوهرية في SDN هي فصل طبقة التحكم عن طبقة البيانات، بحيث تصبح إدارة الشبكة مركزية عبر وحدة تحكم برمجية، بينما يقتصر دور الأجهزة على تمرير البيانات. هذا النموذج يسهّل إدارة الشبكات الكبيرة ويقلل من التعقيد والتكاليف، ويتيح مرونة عالية في ضبط حركة المرور.

قام الباحثون ببناء بيئة تجريبية واقعية تتكون من أربعة أجهزة حاسوب وأربعة مبدلات وجهاز تحكم مركزي، حيث استخدموا أدوات مثل RouteFlow و Mininet و POX للتحكم، بالإضافة إلى خادم Trixbox لتقديم خدمات الاتصال الصوتي والمرئي. في هذه البيئة، تم تشغيل ثلاثة بروتوكولات توجيه مختلفة OSPF : الذي يعتمد على خوارزمية ديكسترا ويختار المسار الأقل تكلفة، BGP (Border Gateway Protocol) المستخدم عادةً بين الأنظمة المستقلة لتبادل المسارات، و RIP البسيط الذي يعتمد على عدد القفزات لتحديد المسار. الهدف كان اختبار أي من هذه البروتوكولات يوفّر أفضل أداء عند تشغيل خدمات الوسائط المتعددة.

لتحقيق ذلك، أجرى الباحثون سلسلة من التجارب العملية، حيث تم قياس جودة الخدمة (QoS) وفق معايير الاتحاد الدولي للاتصالات .(ITU-T) تضمنت القياسات معدل النقل (Throughput)، التأخير (Delay) ، التذبذب(Jitter) ، زمن التقارب (Convergence Time)، وأخيرًا درجة الرأي المتوسط (Mean Opinion Score) MOS التي تعكس رضا

المستخدمين عن جودة الخدمة. النتائج أظهرت أن بروتوكول OSPF كان الأكثر كفاءة بشكل عام، إذ حقق معدل نقل أعلى واستقرارًا أكبر في التأخير والتذبذب، إضافة إلى زمن تقارب أسرع (حوالي 4 ثوانٍ فقط للوصول إلى حالة الاستقرار). أما بروتوكولBGP ، فبالرغم أنه مصمم أساسًا للتعامل مع الشبكات الكبيرة والربط بين الأنظمة المستقلة، فقد أظهر أداءً مقبولًا لكنه تأثر بشكل أوضح عند زيادة الحمل على الشبكة. بينما كان بروتوكول RIP الأقل كفاءة، حيث ارتفعت لديه معدلات التأخير والتذبذب وكان زمن التقارب أطول (حوالي 15 ثانية)، وهو ما يتماشى مع محدوديته المعروفة في الشبكات الكبيرة والمعقدة.

ومن ناحية جودة التجربة للمستخدمين، أظهرت نتائج MOS أن جميع البروتوكولات وفرت جودة مقبولة للمكالمات الصوتية والمرئية، حيث تراوحت القيم حول 4 من أصل 5، وهو تصنيف "جيد" وفق المعايير الدولية. لكن التفوق النسبي لـ OSPF جعله الخيار الأمثل عند تشغيل تطبيقات الوسائط المتعددة، لأنه يوازن بين سرعة الاستجابة واستقرار الأداء حتى مع زيادة الضغط على الشبكة.

وقد خلص الباحثون إلى أن OSPF هو البروتوكول الأكثر ملاءمة، بينما يمكن استخدام BGP و RIP في حالات معينة لكن مع أداء أقل. هذه الدراسة تبرز كيف يمكن للتجارب العملية أن تكشف الفروق الدقيقة بين البروتوكولات المختلفة في سياق التطبيقات الحساسة مثل الصوت والفيديو، لذلك سوف نقوم في بحثنا هذا بتجارب ومحاكاة عملية أكثر لنكشف الفروق بشكل أدق بين البروتوكولات المدروسة.

وأيضاً قام الباحثان (Anusuya et al., 2022) [2] بدراسة كان الهدف الرئيسي منها هو تحليل أداء بروتوكولي التوجيه (OSPF, EIGRP) ضمن الشبكة المدروسة، حيث تناول هذا البحث مقارنة شاملة بين هذين البروتوكولين من حيث الأداء وسرعة الاستجابة للتغيرات في الشبكة، وذلك عبر محاكاة باستخدام برنامج OPNET Modeler وتجارب عملية في المختبر باستخدام أجهزة حقيقية. ينطلق الباحثان من فكرة أن بروتوكولات التوجيه هي الأساس الذي يحدد كيفية انتقال البيانات بين أجهزة الشبكة، وأن اختيار البروتوكول المناسب يؤثر بشكل مباشر على كفاءة الشبكة واستقرارها. لذلك ركزا على هذين البروتوكولين الأكثر شيوعًا في المؤسسات، حيث يعتمد OSPF على خوارزمية ديكسترا (Link State) بينما

يعتمد EIGRP على خوارزمية DUAL (Distance Vector محسّن)، وهو بروتوكول هجين يجمع خصائص من بروتوكولات مختلفة.

في الجانب العملي، صمم الباحثان سيناريوهين في بيئة المحاكاة؛ أحدهما يعمل بـ OSPF والآخر بـ EIGRP، وتمت مراقبة زمن التقارب وعدد الحزم المفقودة. ثم انتقلا إلى التجارب الواقعية في المختبر، حيث بنيا شبكتين مختلفتين: الأولى شبكة اتصال مباشرة (Point-to-Point) لمراقبة سرعة التقارب، والثانية شبكة إيثرنت مشتركة (Shared Ethernet) لمتابعة تأثير عملية انتخاب الموجه الرئيسي والاحتياطي DR/BDR (Designated router/ Backup designated router) في OSPF على زمن التقارب. النتائج أوضحت أن EIGRP يتفوق على OSPF في بيئة الاتصال المباشر، حيث يحقق تقارباً أسرع ويقلل من فقدان الحزم بفضل خوارزمية DUAL (Diffusing Update Algorithm) التي تمنع حدوث الحلقات وتسرّع عملية التحديث. أما في بيئة الإيثرنت المشتركة، فقد كان أداء البروتوكولين متقاربًا، حيث وصلا إلى حالة الاستقرار في وقت واحد تقريبًا، لكن عملية انتخاب DR/BDR في OSPF كانت عاملًا إضافيًا يؤثر على زمن التقارب.

من خلال هذه التجارب، أنجز الباحثان مقارنة عملية بين البروتوكولين وأثبتا أن اختيار البروتوكول يعتمد على طبيعة الشبكة والبنية التحتية المستخدمة؛ فالبروتوكول EIGRP مناسب أكثر للشبكات التي تعتمد على الاتصال المباشر ويتميز بسرعة ومرونة، بينما يظل OSPF بروتوكولًا قياسيًا مفتوحًا مدعومًا على نطاق واسع ويُستخدم بكثرة في بيئات متنوعة. قدمت الدراسة رؤية واضحة للمهندسين وصناع القرار حول كيفية اختيار بروتوكول التوجيه الأنسب بما يتوافق مع متطلبات الشبكة وظروفها العملية، ونحن بدورنا في دراستنا هذه سنقوم بدراسة البروتوكول RIP بالإضافة للبروتوكولين السابقين لنتائج أدق، بالإضافة لإيجاد آلية ديناميكية لاختيار البروتوكول وفق طبيعة الشبكة بشكل آلي، وهو ما يعالجه بحثنا هذا.

ركز البحث الذي قام به الباحثون (Hossain et al., 2020) [3] على مقارنة أداء ثلاثة من بروتوكولات التوجيه الأكثر شيوعًا في الشبكات، وهي EIGRP وOSPF وRIP، وذلك من خلال بيئة محاكاة باستخدام برنامج Cisco Packet Tracer وأداة المحاكاة المتقدمة OPNET. انطلق الباحثون من فكرة أن بروتوكولات التوجيه هي الأساس الذي يحدد كيفية انتقال البيانات بين الأجهزة داخل الشبكة، وأن كل بروتوكول يعتمد على خوارزمية مختلفة في حساب المسارات

وتحديث جداول التوجيه، مما ينعكس بشكل مباشر على سرعة الاستجابة وكفاءة الشبكة. لذلك قاموا بتصميم شبكة موحدة تتكون من عدة أجهزة وموجهات ومبدلات، ثم طبقوا عليها كل بروتوكول على حدة لضمان أن ظروف الاختبار متساوية، بحيث يمكن مقارنة النتائج بشكل عادل.

في الجانب العملي، أنشأ الباحثون سيناريوهات متعددة لكل بروتوكول، حيث تم تشغيل الشبكة باستخدام EIGRP وحده، ثم باستخدام OSPF وحده، وأخيرًا باستخدام RIP، مع مراقبة الأداء في ظروف طبيعية وأخرى تتضمن أعطال متعمدة في الوصلات بين الموجهات لإعادة اختبار قدرة البروتوكولات على استعادة الاستقرار. كما أضافوا تطبيقات واقعية مثل بث الفيديو الخفيف (Video Streaming) لقياس تأثير البروتوكولات على جودة الخدمة في بيئة أقرب إلى الاستخدام الفعلي.

أظهرت النتائج أن بروتوكول EIGRP كان الأسرع في الوصول إلى حالة الاستقرار (Convergence) عند حدوث تغييرات في الشبكة، وذلك بفضل اعتماده على خوارزمية DUAL التي تسمح له بالتعامل مع الأعطال بسرعة ومن دون حدوث حلقات في التوجيه. أما بروتوكول OSPF فقد حقق أداء جيد لكنه كان أبطأ نسبيًا بسبب آلية تحديث قواعد البيانات عبر رسائل LSA التي تتطلب وقتًا أطول لإعادة حساب المسارات. في حين أن بروتوكول RIP كان الأضعف أداءً، حيث استغرق وقتًا أطول في التقارب وأظهر كفاءة أقل في استغلال عرض النطاق الترددي. كما بينت التجارب أن OSPF يولد حجمًا أكبر من حركة المرور الأولية عند بناء صورة كاملة عن الشبكة، بينما كان EIGRP أكثر كفاءة في استهلاك الموارد، وRIP الأقل كفاءة من بين الثلاثة.

من خلال هذه التجارب، أنجز الباحثون مقارنة عملية واضحة بين البروتوكولات الثلاثة، وأثبتوا أن EIGRP هو الأكثر ملاءمة للشبكات التي تحتاج إلى سرعة عالية في الاستجابة واستقرار سريع، بينما يظل OSPF خيارًا قويًا بفضل كونه معيارًا مفتوحًا مدعومًا على نطاق واسع، في حين أن RIP لم يعد مناسبًا للشبكات الكبيرة والمعقدة. الدراسة بذلك تقدم دليلاً عمليًا يساعد المهندسين على اختيار بروتوكول التوجيه الأنسب وفقًا لاحتياجات الشبكة وظروفها، وتؤكد أن المحاكاة باستخدام أدوات مثل Packet Tracer وOPNET يمكن أن توفر رؤية دقيقة لأداء الشبكات قبل تطبيقها في الواقع.

ولقد قام الباحثان (Simson et al.2023) [4] بمقارنة أداء بروتوكولي التوجيه OSPF و EIGRP في بيئة خدمات بث الفيديو عبر الشبكات، وذلك من خلال تحليل جودة الخدمة (QoS) التي تُعدّ معيارًا أساسيًا لقياس كفاءة الشبكات في التعامل مع تطبيقات الوسائط المتعددة.

في سياق الدراسة، انطلق الباحثان من أهمية بروتوكولات التوجيه في ضمان وصول البيانات بشكل صحيح وسريع بين الأجهزة، خاصة مع تزايد الاعتماد على تطبيقات الفيديو التي تتطلب استقرارًا عاليًا وزمن استجابة منخفضًا. لذلك قاما بتصميم شبكة تجريبية يتم فيها تشغيل بروتوكولي OSPF و EIGRP بشكل منفصل، مع مراقبة مؤشرات الأداء الرئيسة مثل معدل التأخير (Delay)، التذبذب (Jitter)، فقدان الحزم(Packet Loss) ، وزمن التقارب (Convergence Time). الهدف كان اختبار أي من البروتوكولين يوفّر تجربة أفضل لمستخدمي خدمات الفيديو عند وجود ضغط على الشبكة أو تغييرات في بنيتها.

أظهرت النتائج أن بروتوكول EIGRP يتميز بسرعة تقارب أعلى، أي أنه يستجيب بسرعة أكبر للتغيرات في الشبكة ويعيد الاستقرار خلال وقت أقصر، وهو ما ينعكس إيجابًا على استمرارية بث الفيديو دون انقطاع ملحوظ. كما أظهر أداءً أفضل في تقليل التأخير والتذبذب، مما يجعل تجربة المشاهدة أكثر سلاسة. في المقابل، قدّم بروتوكول OSPF أداءً جيدًا أيضًا، لكنه كان أبطأ نسبيًا في الوصول إلى حالة الاستقرار بسبب آلية تحديث قواعد البيانات عبر رسائل LSA (Link State Advertisement) التي تتطلب وقتًا أطول لإعادة حساب المسارات. ومع ذلك، يظل OSPF خيارًا قويًا بفضل كونه معيارًا مفتوحًا مدعومًا على نطاق واسع، بينما يقتصر استخدام EIGRP على بيئات تعتمد على أجهزة سيسكو.

من خلال هذه التجربة، أنجز الباحثان مقارنة عملية واضحة بين البروتوكولين في سياق خدمات الفيديو، وأثبتا أن EIGRP أكثر ملاءمة للشبكات التي تحتاج إلى سرعة عالية في الاستجابة واستقرار سريع، بينما يظل OSPF خيارًا مناسبًا للشبكات الكبيرة والمعقدة التي تتطلب معيارًا مفتوحًا ومرونة في التوسع. الدراسة بذلك تقدّم دليلًا عمليًا يساعد المهندسين وصنّاع القرار على اختيار بروتوكول التوجيه الأنسب وفقًا لاحتياجات الشبكة وظروفها، وتؤكد أن جودة الخدمة في تطبيقات الفيديو

تعتمد بشكل مباشر على كفاءة البروتوكول المستخدم في إدارة حركة المرور، وهذا ما سندرسه ونتوسع بدراسته في هذا البحث.

أما الباحث (Biradar.2020) [5] فقد قام بدراسة مقارنة بين بروتوكولات التوجيه الثلاثة الأكثر شيوعًا في الشبكات، وهي RIP وOSPF وEIGRP، وذلك باستخدام بيئة المحاكاة العملية GNS-3 التي تتيح بناء شبكات افتراضية أقرب إلى الواقع. ينطلق الباحث من أن عملية التوجيه هي جوهر عمل الإنترنت، وأن البروتوكولات المختلفة تعتمد على خوارزميات متباينة لتحديد أفضل المسارات لنقل البيانات، مما ينعكس بشكل مباشر على الأداء العام للشبكة. لذلك صمّم بيرادار شبكة تجريبية مكوّنة من مجموعة من أجهزة التوجيه والمبدلات والحواسيب الافتراضية، ثم طبّق عليها كل بروتوكول على حدة، مع مراقبة مؤشرات الأداء الأساسية مثل زمن التأخير (Delay) وزمن التقارب (Convergence Time).

في التجارب، أظهر بروتوكول RIP بساطته وسهولة تنفيذه، لكنه كان الأقل كفاءة، حيث سجّل أعلى زمن تأخير وأطول زمن تقارب، وهو ما يعكس محدوديته في الشبكات الكبيرة والمعقدة. أما بروتوكول OSPF فقد حقق نتائج أفضل من RIP، إذ سجّل أقل زمن تأخير بفضل اعتماده على خوارزمية ديكسترا التي تسمح له باختيار المسار الأقل تكلفة بسرعة، لكنه كان أبطأ نسبيًا في زمن التقارب بسبب آلية تحديث قواعد البيانات عبر رسائل الـ LSA التي تتطلب وقتًا أطول لإعادة حساب المسارات عند حدوث تغييرات. في المقابل، أثبت بروتوكول EIGRP تفوقًا واضحًا، حيث سجّل أسرع زمن تقارب (حوالي 3 ثوانٍ فقط) بفضل اعتماده على خوارزمية DUAL التي تتيح له الاحتفاظ بمسار بديل جاهز (Feasible Successor) يمكن استخدامه فورًا عند فشل المسار الرئيسي، مما يقلل من فقدان الحزم ويعيد الاستقرار بسرعة. كما أظهر أداءً جيدًا في زمن التأخير، ليجمع بين سرعة الاستجابة وكفاءة الأداء.

من خلال هذه الدراسة، أنجز الباحث مقارنة عملية دقيقة بين البروتوكولات الثلاثة، وأظهرت نتائج المحاكاة أن البروتوكولين EIGRP وOSPF يتمتعان بزمن تأخير أفضل مقارنةً بـ RIP. وأظهرت نتائج المحاكاة أيضاً أن EIGRP يتمتع بأقل زمن تقارب مقارنةً بـ OSPF وRIP وخلص الباحث أنه بالنسبة لهذه الشبكة المدروسة فإن البروتوكول EIGRP هو أفضل بروتوكول توجيه على الشبكة المدروسة.

وأيضاً قام الباحثون (Tsochev et al.2021) [6] بمقارنة بروتوكولي التوجيه OSPF وEIGRP من خلال المحاكاة باستخدام برنامج OPNET Modeler، وذلك بهدف تقييم أدائهما في بيئات شبكية واقعية تعتمد على تطبيقات حساسة للزمن مثل الاتصالات الصوتية والفيديو والمؤتمرات عبر الإنترنت. ينطلق الباحثون من أن بروتوكولات التوجيه هي العمود الفقري للشبكات الحديثة، فهي التي تحدد كيفية تواصل الموجّهات فيما بينها، وتبادل المعلومات، وبناء جداول التوجيه، واختيار أفضل المسارات لنقل البيانات. لكن لكل بروتوكول خصائصه وخوارزمياته المختلفة، مما يجعل أداءه متباينًا بحسب حجم الشبكة وطبيعتها.

قام الفريق أولًا بدراسة نظرية موسّعة لخصائص كل بروتوكول، وبعد ذلك صمّم الباحثون بيئة محاكاة تتكون من عدة شبكات فرعية موزعة على مدن مختلفة تحاكي البنية الجغرافية لشبكة وطنية، حيث تم تشغيل البروتوكولين بشكل منفصل على نفس البنية لضمان عدالة المقارنة. أضيفت تطبيقات واقعية مثل مؤتمرات الفيديو عالية الدقة، الاتصالات الصوتية عبر بروتوكول الإنترنت، وتصفح الويب الكثيف، كما تم إدخال أعطال متعمدة في الوصلات بين الشبكات لإعادة اختبار قدرة البروتوكولات على الاستجابة واستعادة الاستقرار.

من خلال هذه التجارب، جمع الباحثون بيانات كمية حول مؤشرات الأداء مثل تباين تأخير الحزمة، وتأخير الحزمة من طرف إلى طرف، وحركة المرور المستلمة، وحركة المرور المرسلة.

وقد كان تباين تأخير الحزمة متشابه جداً بالنسبة للبروتوكولين السابقين، ولكن EIGRP يعكس متوسط تأخير أعلى قليلاً، وبالتالي، يتمتع بإنتاجية أقل مقارنةً بـ OSPF. وفي حالة التأخير من طرف إلى طرف، لم تتم ملاحظة أي اختلاف كبير بين أداء OSPF وEIGRP. وأشار الباحثون إلى وجود العديد من العوامل التي تلعب دوراً حاسماً في اختيار البروتوكول الذي سيتم استخدامه في كل حالة، مثل البنية التحتية وحجم الشبكة ونوع المتطلبات التي يجب تلبيتها في كل مرة، وهذا ما سنتناوله في دراستنا هذه.

أيضاً قام الباحثان (Noelia et al. 2023) [7] بتقييم أداء البروتوكولين OSPF وEIGRP ضمن نوعين من طوبولوجيا الشبكات عند تشغيل خدمتي الفيديو والصوت بهدف إجراء تحليل مقارن لأداء هذين البروتوكولين من خلال بيئة محاكاة

باستخدام أدوات مثل Cisco Packet Tracer، مع التركيز على مؤشرات كمية دقيقة مرتبطة بجودة الخدمة في التطبيقات الزمنية الحساسة.

انطلق الباحثان من أن بروتوكولات التوجيه الديناميكية هي الأساس في عمل الشبكات الحديثة، وأن اختيار البروتوكول المناسب يؤثر مباشرة على الأداء العام للشبكة، خاصة في تطبيقات مثل مؤتمرات الفيديو والمكالمات الصوتية عبر بروتوكول الإنترنت (VoIP). لذلك قاما بتصميم سيناريوهات شبكية متشابهة، بحيث يتم تشغيل البروتوكولين بشكل منفصل على نفس البنية، لضمان عدالة المقارنة.

المؤشرات التي تم اعتمادها في الدراسة شملت زمن التقارب (Network Convergence Time)، وتغير التأخير في مكالمات الفيديو (Variation of Packet Delay in a Video Conference Call)، وتذبذب الصوت (IP Voice Jitter)، إضافة إلى استهلاك وحدة المعالجة المركزية (CPU Utilization). هذه المعايير تم قياسها بدقة في بيئة المحاكاة، حيث أُنشئت شبكات افتراضية تحاكي ظروف الاستخدام الواقعي، وتم إدخال تغييرات في البنية الشبكية لمراقبة كيفية استجابة كل بروتوكول.

النتائج التي جمعها الباحثان أوضحت الفروق بين البروتوكولين في هذه المؤشرات، حيث تم تسجيل بيانات كمية دقيقة لكل حالة، وتم تحليلها بشكل مقارن لإبراز نقاط القوة والضعف لكل بروتوكول. الدراسة لم تقتصر على الجانب العملي فقط، بل تضمنت أيضًا مراجعة نظرية لخصائص كل بروتوكول، حيث تم شرح آلية عمل OSPF باعتباره بروتوكول حالة ارتباط يعتمد على خوارزمية ديكسترا، وEIGRP باعتباره بروتوكولًا هجينًا يعتمد على خوارزمية DUAL ويجمع بين خصائص المتجه والمسار.

في النهاية، أنجز الباحثان تقييمًا شاملًا لأداء البروتوكولين في سياق التطبيقات الزمنية الحساسة، وكانت النتيجة أن البروتوكول EIGRP يقدم أداءً فائقاً، خاصة فيما يتعلق بزمن تقارب الشبكة واستهلاك موارد وحدة المعالجة المركزية، وأوصى الباحثان أنه ينبغي دراسة طوبولوجيا الشبكات الأكثر تعقيداً من أجل نتائج أكثر دقة وهذا أيضاً ما سنعمل على أخذه بعين الاعتبار في هذه الدراسة.

أما الدراسة التي قام بها (Shahid et al. 2024) [8] فقد سلطت الضوء على مزايا البروتوكول EIGRP مقارنةً بالبروتوكول OSPF ولا سيما من حيث فقدان حزم البيانات وزمن تقارب فشل الوصلة، في بيئة شبكية تعتمد على بروتوكول IPv6، مع التركيز على أنظمة مشاركة الحمل (Load Sharing) وتجاوز الأعطال (Link Failover).

صمم الباحثون آلية جديدة تسمح باستخدام الوصلات الأساسية والاحتياطية بشكل متزامن، مع توزيع الحمل بينهما، وإدارة نوعي الحركة عبر شبكتين منفصلتين: شبكة Overlay آمنة لحركة البيانات الداخلية، وشبكة Underlay لحركة الإنترنت، مع تطوير آلية لتوجيه المسارات تدعم المشاركة في الحمل وتجاوز الأعطال وفق الحاجة.

لتحقيق ذلك، استخدم الباحثون بيئة محاكاة متقدمة تضمنت GNS3 لمحاكاة الطوبولوجيا الشبكية، وVMware للبيئة الافتراضية، وWireshark لتحليل الحزم على مستوى البروتوكول، تم اختبار البروتوكولات الثلاثة في ظروف مختلفة تشمل تغيّر أحمال الشبكة وحدوث أعطال في الوصلات، وتم قياس مؤشرات مثل زمن التقارب (Convergence Time)، فقدان الحزم (Packet Loss)، التأخير (Delay)، والتذبذب (Jitter).

النتائج التي حصل عليها الباحثون أوضحت أن بروتوكول EIGRP قدّم أداءً أفضل من حيث زمن التقارب عند حدوث الأعطال، وكذلك نسبة فقدان الحزم، مقارنةً بـ OSPF، بينما أظهر OSPF استقرارًا جيدًا في إدارة المسارات لكنه كان أبطأ في إعادة الحساب عند تغيّر البنية.

بهذا، أنجز الباحثون دراسة عملية دقيقة قدّمت آلية جديدة لتفعيل الوصلات الاحتياطية بشكل فعّال، وتوفير رؤية شاملة لأداء الشبكات في سيناريوهات المشاركة في الحمل وتجاوز الأعطال، مع تقديم بيانات كمية واضحة تساعد المهندسين على اختيار البروتوكول الأنسب لتحسين الأداء وضمان استمرارية الخدمة.

فبينما أظهر كلا البروتوكولين اختلافات طفيفة في زمن وصول الشبكة وتذبذبها، فقد أظهر البروتوكول EIGRP باستمرار أداءً متفوقاً في سيناريوهات تجاوز فشل الوصلة، سواءً تم تفعيل المصادقة أم لا، حيث تعافى بشكل أسرع من البروتوكول OSPF. بالإضافة إلى ذلك، يشير انخفاض معدل فقدان الحزم الملحوظ في البروتوكول EIGRP إلى قدرته الأفضل على التعامل مع تحديات الشبكة بشكل أفضل، مما يُسهم في نقل بيانات أكثر موثوقية. وتتجلى أهمية اختيار بروتوكول

توجيه مناسب في التأثير الكبير الذي يُمكن أن تُحدثه حتى الاختلافات الطفيفة في تطبيق مقاييس الأداء على الأداء العام للشبكة وموثوقيتها. قدمت هذه الدراسة نهجاً شاملاً، متجاوزةً سيناريوهات فشل الوصلات التقليدية، لتشمل استراتيجيةً أكثر شموليةً تتناول كلاً من فشل الاتصال ومشاركة الحمل عبر أنواع حركة مرور مختلفة، وهو ما سنقوم به أيضاً بدراستنا هذه ولكن بواسطة تطبيق الخوارزمية المقترحة المطورة من قبلنا على نفس الشبكة مع تطبيق بروتوكولي التوجيه OSPF و EIGRP ومقارنة النتائج بعد تطبيق الخوارزمية المطورة على الشبكة.

أما فيما يخص دراسة خوارزمية مستعمرة النمل وتعديلها في الشبكات فكان هنالك العديد من الدراسات التي تتناول المسار الأقصر في الشبكات في حين لم يكن هناك دراسات خاصة بها تستخدم بروتوكولات التوجيه أو إدارتها حسب أنواع البيانات، فقد قام الباحثون (Ameen et al. 2016) [9] بتحديد مسارات التوجيه في الشبكة، وتعديلها من أجل تحديد أفضل مسار في حالة الازدحام وأظهرت النتائج أن هذه الطريقة تسمح للشبكة بتنشيط مسار جديد إذا كان هناك توقف للخدمة أو ازدحام أو عنق زجاجة في المسار الحالي بالإضافة لتحديد أقصر مسار. وقد تناولت مشكلة التوجيه في الشبكات الحديثة، حيث أوضح الباحثون أن النمو السريع لشبكات الاتصال يفرض تحديات كبيرة في إيجاد المسارات المثلى لنقل البيانات، خاصة في الشبكات ذات التبديل بالحزم. أشاروا إلى أن إضافة عقد جديدة أو حدوث أعطال في الوصلات يؤدي إلى اضطراب في عمل أجهزة التوجيه، مما يستلزم إعادة حساب المسارات الأقصر، مع ضرورة مراعاة عوامل مثل التحكم في الازدحام وتقليل التأخير لضمان استمرار تدفق البيانات دون فقدان.

لحل هذه المشكلات، اعتمد الباحثون على تقنيات الذكاء الاصطناعي، وبالتحديد خوارزمية تحسين مستعمرة النمل (Ant Colony Optimization – ACO)، وقاموا بتطويرها لتناسب بيئة التوجيه الشبكي الديناميكي. الفكرة الأساسية تقوم على محاكاة سلوك النمل في البحث عن الغذاء، حيث يتم إرسال "نمل اصطناعي" يمثل الحزم عبر الشبكة، ويقوم بترك أثر الفيرومون على المسارات التي يسلكها. كلما كان المسار أقصر وأكثر نجاحاً، زادت قيمة الفيرومون عليه، مما يوجه الحزم التالية لاختيار هذا المسار. الباحثون صاغوا معادلات رياضية لتحديث قيم الفيرومون بشكل دوري، بحيث يتم تعزيز المسارات الجيدة وتقليل قيمة المسارات الأقل كفاءة.

تم تنفيذ الخوارزمية باستخدام لغة Visual Basic.Net 2012، وأُجريت تجارب عملية على شبكة مكونة من 20 عقدة. في الاختبار الأول، تمكنت الخوارزمية من إيجاد أقصر مسار بطول 1971 متر من أصل طول إجمالي للشبكة بلغ 48733 متر. في الاختبار الثاني، بعد إزالة عقدتين ليصبح عدد العقد 18، حددت الخوارزمية مسارًا أقصر بطول 1612 متر من أصل 33289 متر. أما في الاختبار الثالث، ومع إزالة عقدتين إضافيتين ليصبح العدد 16، فقد حددت الخوارزمية مساراً بطول 1005 متر من أصل 20015 متر. هذه النتائج أثبتت أن الخوارزمية قادرة على التكيف مع تغيّر حجم الشبكة، وأنها فعّالة في إيجاد المسارات الأقصر حتى عند حدوث أعطال أو تغييرات في البنية.

خلص الباحثون إلى أن استخدام تقنيات الذكاء الاصطناعي، وخاصة تحسين مستعمرة النمل، يوفر حلاً عملياً لمشكلات التوجيه في الشبكات، حيث يسمح بتحديث جداول التوجيه بشكل ديناميكي، وتحديد أفضل مسار في حالة الازدحام أو انقطاع الوصلات. وأكدوا أن الخوارزمية المقترحة تجعل الشبكة أكثر تكيفاً وكفاءة وقابلية للتوسع، وأوصوا في المستقبل بتطبيقها مباشرة داخل أجهزة التوجيه في الشبكات الحقيقية للتحقق من فعاليتها في بيئات تشغيل واقعية، ونحن بدورنا سنعمل على إيجاد حلول للتوجيه ولكن حسب نوع البيانات المنقولة ضمن الشبكة وأيضاً معالجة مشاكل انقطاع الوصلات ضمن هذا البحث وبأفضل جودة ممكنة.

كما قام الباحثون (Mamta et al. 2016) [10] بدراسة مسارات التوجيه في الشبكة، حيث تناولت الدراسة خوارزمية تحسين مستعمرة النمل (Ant Colony Optimization – ACO) وتطبيقاتها في بروتوكولات التوجيه، مع مقارنة دقيقة بينها وبين البروتوكولات التقليدية مثل RIP وOSPF. أوضحت الباحثتان أن خوارزمية ACO تُعد إطارًا متعدد الوكلاء لحل مشكلات التحسين التوافقي، حيث يقوم "النمل الاصطناعي" ببناء الحلول تدريجيًا عبر إضافة مكونات واحدة تلو الأخرى، وهي مناسبة بشكل خاص للشبكات الديناميكية التي تتغير بنيتها باستمرار، إذ يمكن تشغيلها بشكل متواصل والتكيف مع التغييرات في الزمن الحقيقي.

في القسم الخاص بمعلومات التوجيه، شرحت المقالة كيف أن بروتوكول RIP يعتمد على تبادل جداول المسافة بين العقد، بينما يحتاج OSPF إلى معرفة حالة الوصلات كاملة لحساب أقصر مسار، في حين أن ACO يستكشف المسارات من

المصدر إلى الوجهة بشكل متوازي ومستقل باستخدام عدد من النمل، حيث يتم تحديث قيم الفيرومون على المسارات فور وصول النمل إلى العقد، مما يسمح بتحديث جداول التوجيه بشكل مستقل وسريع.

أما في جانب الحمل الزائد للمعلومات، فقد بيّنت الدراسة أن RIP يتطلب إرسال جداول توجيه كاملة بشكل دوري، وهو ما يسبب حملًا كبيرًا في الشبكات الواسعة، بينما يعتمد OSPF على إرسال رسائل حالة الوصلات (LSP) عبر عملية فيض تؤدي إلى تكرار نسخ متعددة من نفس الرسالة، وهو ما يزيد من الحمل أيضًا. في المقابل، يعتمد ACO على إرسال نمل صغير الحجم (حوالي 6 بايت)، مما يقلل بشكل كبير من الحمل الزائد ويجعل التحديثات أكثر كفاءة.

في جانب القدرة على التكيف، أوضحت الدراسة أن البروتوكولات التقليدية قد تواجه بطئاً في الاستجابة عند حدوث تغييرات في الطوبولوجيا إذا طالت الفترات بين التحديثات، بينما يمكن للنمل في ACO أن يُدمج مع الحزم المرسلة ويُرسل بشكل متكرر لتحديث المعلومات بسرعة، مما يجعله مناسبًا للشبكات الديناميكية. كما ناقشت مشكلة الركود (Stagnation) التي قد تحدث عندما يتركز الفيرومون على مسار واحد بشكل مفرط، مما يؤدي إلى ازدحام ذلك المسار أو فقدانه عند حدوث عطل، فقد يحدث الركود في الشبكة عندما يتم اختيار مسار ما من قبل جميع النمل والذي يزيد بشكل متكرر بسبب تفضيلها له مما قد يؤدي بعد ذلك إلى: ازدحام، وانخفاض كبير في احتمال اختيار مسارات أخرى. وهذان الأمران غير مرغوب فيهما لشبكة ديناميكية نظراً لما يلي:

- قد تصبح غير مثالية إذا كانت مزدحمة.
- قد يتم قطع الاتصال بسبب فشل الشبكة.
- قد تصبح المسارات الأخرى غير المثالية هي الأمثل بسبب التغيرات في طوبولوجيا الشبكة.
- قد يتم اكتشاف مسارات جديدة أو أفضل.

أما في دراستنا هذه، فسوف يتم تطوير خوارزمية تحسين مستعمرة النمل (ACO) الأمر الذي سوف يسمح باختيار المسارات المؤدية للوجهة ومن ثم تشغيل بروتوكولات التوجيه المناسبة عليها بشكل آلي حيث أنه بفشل مسار معين يقوم

بروتوكول التوجيه فيما بعد باختيار مسار آخر وبشكل ديناميكي، فينتج عن ذلك الدمج الاستفادة القصوى من قوة بروتوكولات التوجيه التقليدية مع مرونة خوارزميات الذكاء الصنعي للحصول على أفضل أداء للشبكة عند تقديم الخدمات المتعلقة بها.

الفصل الثالث

أساسيات خوارزمية تحسين مستعمرة النمل

Basics of Ant Colony Optimization Algorithm

1.3. مقدمة عن خوارزمية تحسين مستعمرة النمل (ACO) :

خوارزمية مستعمرات النمل (ACO) هي طريقة ذكية مستوحاة من الطبيعة لحل مسائل معقدة (خصوصاً مسائل البحث عن أفضل مسار أو التحسين التوافقي). الفكرة جاءت من سلوك النمل في الطبيعة :عندما يخرج النمل للبحث عن طعام، تبدأ كل نملة بشكل عشوائي، ولكن أثناء السير تترك خلفها مادة كيميائية تُسمى الفيرمون، كلما زاد عدد النمل المار على مسار معين، تراكم الفيرمون عليه، فيزداد احتمال أن يتبعه نمل آخر، ومع الوقت، يظهر أن المسار الأقصر أو الأفضل يحصل على فيرومون أكثر لأنه يُستخدم أكثر، وبالتالي تصبح المستعمرة كلها تقريباً تسلك هذا المسار.

جاءت تلك التسمية لأن الخوارزمية تحاكي سلوك مستعمرة النمل الحقيقية في الطبيعة لحل مسائل التحسين (Optimization problems) أي كما يكتشف النمل المسارات القصيرة للغذاء، تستعمل الخوارزمية نملاً اصطناعياً (برمجياً) لاكتشاف الحلول المثلى لمسائل رياضية.

ظهرت الفكرة أول مرة في أوائل التسعينيات مع الباحث الإيطالي Marco Dorigo في أطروحته للدكتوراه، وسُمِّيت أول نسخة من الخوارزمية Ant System وكانت تستهدف حل مسألة البائع المتجول (Travelling Salesman Problem – TSP)، وهي من أصعب مسائل الأمثلة على التحسين التوافقي.

لاحقاً، طُوِّرت نسخ أخرى أكثر فعالية مثل: Ant Colony System (ACS) و MAX–MIN Ant System (MMAS).

2.3. خوارزمية تحسين مستعمرة النمل (ACO):

تعتمد الخوارزمية على مجموعة من العوامل البسيطة (نمل اصطناعي) تتعاون عبر اتصال غير مباشر بوساطة الفيرمونات لبناء حلول لمسائل التحسين التوافقي الصعبة. الحلول الجيدة تظهر كسلوك ناشئ من تفاعل هذه العوامل، ويُخصَّص الجهد الحسابي لعدد كبير من العوامل بدل وكيل واحد معقّد. حيث يمكن استخدام ACO لمسائل ساكنة أو ديناميكية.

فكرة ACO الأساسية : تعتمد خوارزمية تحسين مستعمرة النمل (ACO) في جوهرها على استخدام قيم الفيرمون (τ) كتمثيل احتمالي لتوجيه عملية توليد الحلول، ويتم تحديث هذه القيم تدريجياً بحيث تزداد احتمالية إنتاج حلول ذات جودة عالية مع مرور الوقت. يُنظر إلى هذا النهج على أنه يتم استكشاف فضاء معاملات التوزيع الاحتمالي بهدف زيادة فرص الوصول إلى حلول مثلى.

1.2.3. مكوّنات النموذج لتطبيق مشكلة البائع المتجول TSP (Travelling salesman problem):

- **الرسم البنائي (Construction Graph):** العقد (رؤوس)، والانتقالات (الحواف) بين العقد.
- **أثر الفيرمون τ_{ij}:** يرمز إلى جاذبية اختيار القوس (i,j)
- **المعلومة الإرشادية (Heuristic):** $\eta_{ij}=1/d_{ij}$ عكس المسافة، وتُستعمل كتحيز إرشادي مسبق .
- **ذاكرة كل نملة M_k:** قائمة العقد المزارة لترسيم الجوار المسموح N_i^k (العقد غير المزارة بعد من العقدة i)، ولحساب طول جولتها وإعادة تتبع المسار عند الإيداع .
- **تمثيل القرار وحسابات السرعة:** تُخزَّن القيم $[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}$ في مصفوفة معلومات الاختيار(choice_info) لتسريع اختيار الانتقال، مع طرق عملية لتفادي إعادة الحساب .

تبني النملة جولتها خطوة بخطوة وعند تموضع النملة k في العقدة i، فإن احتمال اختيار عقدة مجاورة j (غير مزارة بعد) يُعطى بـ المعادلة:

$$p_{ij}^k = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in \mathcal{N}_i^k}[\tau_{il}]^{\alpha}[\eta_{il}]^{\beta}}, \quad \text{if } j \in \mathcal{N}_i^k \qquad [11]$$

حيث $\eta_{ij}=1/d_{ij}$، و α,β يوازنان تأثير الفيرمون مقابل المعلومة الإرشادية، فإذا كان $\alpha = 0$ فإن العقد الأقرب تصبح أكثر احتمالاً للاختيار، وهو ما يُحاكي الخوارزمية الجشعة (Greedy Algorithm)، حيث تُستخدم نقاط انطلاق متعددة

نظراً لتوزّع النمل عشوائياً على العقد في بداية التنفيذ. ؛ وإذا كان $\beta = 0$ اعتمدنا على الفيرمون وحده فذلك يعطي نتائج ضعيفة وقد يقود لركود مبكر عند قيم $\alpha > 1$ بسبب التضخيم الراجع.

تهيئة الفيرمون كما يلي:

$$\tau 0 = m / C_{nn}$$

حيث τ_0 هي قيمة تركيز الفيرمون الابتدائية، و m عدد النمل، و C_{nn} طول جولة مُنشأة بخوارزمية الجار الأقرب.
السبب: إن كانت τ_0 صغيرة جدّاً يتحيز البحث مبكراً لجولات أولية رديئة؛ وإن كانت كبيرة جدّاً يضيع الزمن قبل أن يؤدي التبخر إلى مستويات يُمكن فيها لإيداعات النمل أن تُوجِّه البحث. [11]

3.3. تحديث الفيرمون:

يتم تحديث الفيرمون الكلي بعد إكمال جميع النمل للجولات بالإضافة لما يلي:

- **التبخّر (لتفادي تراكم غير محدود ولتجاهل القرارات السيئة):**

$$(1 - \rho)\ \tau_{ij} \longrightarrow \tau_{ij}\ ,\ \forall (i,j) \in L$$

$$0 < \rho \leq 1$$

- **الإيداع على الحواف التي استُخدمت في جولات النمل، بمقادير تتناسب مع جودة الجولة (أقصر = أفضل):**

$$\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \longrightarrow \tau_{ij}\ ,\forall (i,j) \in L$$

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{C_k}, & \text{إذا كان القوس } (i,j) \text{ ضمن جولة النملة } k \\ 0, & \text{غير ذلك} \end{cases}$$

تعبر C_k هنا عن طول جولة النملة k، حيث أن جولات أقصر تُكافأ بإيداع أكبر، ومع التبخّر ينخفض أثر الحواف غير المستخدمة أُسّياً عبر التكرارات.

4.3. خواص وسلوك الخوارزمية ACO:

تمتاز خوارزمية ACO بالعديد من الميزات والتي تتركز فيما يلي:

1.4.3. تغذية راجعة إيجابية + تبخر = توازن استكشاف/استغلال: يودِع النمل الفيرمون على المسارات الجيدة، فتجذب المزيد من النمل إليها (تضخيمٌ ذاتي/استجابة إيجابية)، وهو مبدأ مستوحى من تجارب "الجسر المزدوج" على النمل الحقيقي. لكن التبخر ضروري هنا لضبط التراكم والحد من الانجراف نحو مساراتٍ دون المثلى.

2.4.3. تأثير المعاملات α,β والركود: إذا كانت قيم α كبيرة فإنها تُعجّل بالركود (جميع النمل يتبع نفس المسار سريعاً)، بينما غياب β يُضعف الجودة. تم اقتراح قيماً معقولة لتفادي استكشافٍ مفرط أو استغلالٍ مفرط، وتم وضع مؤشراتٍ كميةً لرصد الركود، مثل متوسط عامل التفرّع λ أو إنتروبيا التوزيعات الاحتمالية حيث تؤدي الإعدادات السيئة إلى تدهور أو ركود مبكّر.

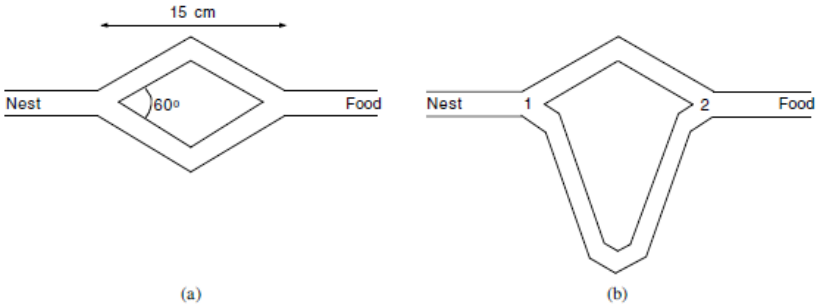
3.4.3. عدد النمل وقوائم الجوار المرشحة: عادةً يُتَّخذ $m= n$ دون بحثٍ محلي. لتطبيقاتٍ كبيرة، قوائم الجوار تُقلّص كلفة القرار كثيراً وغالباً ما تحسّن السرعة دون الإضرار بالجودة.

العناصر/الأدوار داخل الخوارزمية:

- النمل (وكلاء بسيطون) : يتحرك بقواعد محلية، تقرأ τ ، وتبني حلاً جزئياً مع ذاكرة قصيرة لمنع التراجع الفوري والحلقات .
- مسارات الفيرمون τ : معاملات نموذج احتمالي تعبّر عن «قابلية» اختيار مكوّنات الحل؛ تُحدَّث بعد كل جولة وفق جودة الحلول.

- التبخّر ρ (فرامل) لمنع التكلّس على مسارات رديئة مبكراً، ويزيد تنوّع البحث .
- الاستدلال الإرشادي (اختياري في الإطار العام): معلومات مثل $1/d$ في TSP لتعزيز المسارات القريبة، تُدمج عادةً مع τ في قاعدة الاختيار (في نسخ ACO العملية) .
- لماذا تُعد ACO مفيدة في التطبيق؟
  - بساطة محليّة + ذكاء جماعي: وكلاء بسطاء ينتجون سلوكاً جماعياً قوياً .
  - استغلال/استكشاف متوازن: التعزيز يركّز على الجيّد، والتبخّر يُبقي الباب مفتوحاً للجديد .
  - تُستخدم الخوارزمية في حل مسائل صعبة، وذلك من خلال تمثيل مناسب للمشكلة وتعريف واضح لمعيار الجودة، مع إمكانية دمج تحسينات محلية فعالة.

4.4.3. الأساس البيولوجي (تجارب الجسرين):



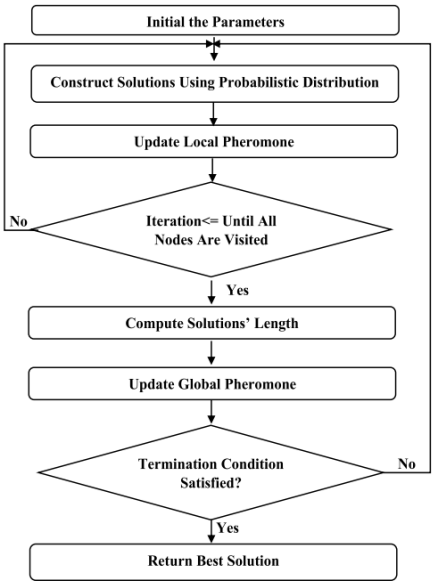
الشكل (3–1)– تجربة الجسرين [11]

أظهرت تجارب (الجسر المزدوج) أنّ النمل يختار تلقائياً المسار الأقصر عبر الفيرمون مع تغذية راجعة إيجابية كلما ازداد المرور على فرع ازدادت كثافة الفيرمون فيزيد ذلك احتمال اختياره لاحقاً، فتتجه المستعمرة إليه كلها. هذه الظاهرة صيغت بعدة نماذج احتمالية ومعادلات تفاضلية أظهرت كيف يقود الفرق في الزمن (الوصول الأسرع عبر المسار القصير) إلى تراكم أسرع للفيرمون هناك [11] .

ويمكن تمثيل خوارزمية تحسين مستعمرة النمل (ACO) بالخطوات التالية وكما هو موضح بالشكل (3–2):

5.3. خطوات خوارزمية (ACO) الأساسية:

- تهيئة الإعدادات: تحديد عدد النمل، كمية الفيرمون، وعدد التكرارات.
- بناء الحلول: كل نملة تختار طريقها بناءً على الفيرمون والمسافة.
- تحديث الفيرمون المحلي: كل نملة تترك أثراً مؤقتاً على الطريق الذي سلكته.
- تكرار حتى زيارة كل النقاط: تستمر النمل في بناء الحلول حتى تغطي كل العقد.
- حساب طول الحلول: قياس جودة كل طريق.
- تحديث الفيرمون العام: تعزيز الطرق الأفضل بكمية أكبر من الفيرمون.
- التحقق من شرط الإنهاء: إذا لم يتحقق، تبدأ دورة جديدة.
- إرجاع أفضل حل: يتم اختيار أفضل طريق تم اكتشافه.



الشكل (3-2)- مخطط خوارزمية ACO [12]

الفصل الرابع

خوارزمية البحث بالعمق أولاً

Depth-first search algorithm (DFS)

1.4. مقدمة لخوارزمية البحث بالعمق أولاً:

تُعد خوارزمية البحث بالعمق من أبرز الخوارزميات المستخدمة في مجال هياكل البيانات والخوارزميات، وهي حجر أساس في فهم كيفية استكشاف البيانات المترابطة مثل الأشجار والرسوم البيانية (Graphs). تنتمي هذه الخوارزمية إلى فئة ما يُعرف بخوارزميات البحث الأعمى، أي تلك التي لا تعتمد على معلومات مسبقة عن الهدف، بل تستكشف المسارات بشكل منهجي حتى تصل إلى الحل أو تستنفد جميع الاحتمالات.

ما يميز DFS هو طريقتها في الاستكشاف، فهي لا تكتفي بالبحث السطحي أو الأفقي، بل تغوص في أعماق البنية، وتتابع كل مسار حتى نهايته قبل أن تعود وتبحث في مسار آخر. هذا النهج يجعلها مثالية في الحالات التي يكون فيها الحل موجوداً في أعماق البنية، أو عندما نرغب في استكشاف جميع الاحتمالات الممكنة.

تُستخدم DFS في مجموعة واسعة من التطبيقات، بدءاً من حل الألغاز والمتاهات، مروراً بتحليل الشبكات الاجتماعية، ووصولاً إلى الذكاء الاصطناعي وألعاب الفيديو. كما تلعب دوراً مهماً في اكتشاف الدورات داخل الرسوم البيانية، وتحديد المكونات المتصلة، وتوليد الترتيبات الممكنة للعناصر.

2.4. آلية العمل:

تتمثل استراتيجية البحث بالعمق أولاً، كما يوحي اسمها، في الغوص بأعماق الرسم البياني كلما سنحت الفرصة، حيث يستكشف البحث بالعمق أولاً الحواف المتصلة بأحدث رأس مُكتشف والذي لا تزال به حواف غير مُكتشفة بعد، وبمجرد استكشاف جميع الحواف، تعود الخوارزمية إلى الرأس السابق الذي تم الوصول إليه، وتتابع الاستكشاف انطلاقاً منه. تستمر هذه العملية حتى يتم اكتشاف جميع الرؤوس التي يمكن الوصول إليها من الرأس المصدر الأصلي. وإذا تبقت رؤوس غير مكتشفة، تختار الخوارزمية أحدها كمصدر جديد وتعيد تنفيذ العملية ذاتها، حتى يتم اكتشاف جميع الرؤوس في الرسم البياني.

وعندما تكتشف خوارزمية البحث بالعمق رأساً جديداً أثناء فحص الرؤوس المجاورة لرأس تم اكتشافه سابقاً u، فإنها تسجّل هذا الاكتشاف بربط الرأس الجديد بالرأس الذي قاد إلى اكتشافه وتعيين سمة السلف إلى u ، وتعتبره "السلف" أو السابق له. ومع تكرار هذه العملية، تتكوّن بنية تُظهر كيف تم الوصول إلى كل رأس. وعلى عكس البحث بالاتساع الذي يُنتج شجرة واحدة متصلة، فإن البحث بالعمق قد يُنتج عدة أشجار منفصلة.

يُشكل الرسم الفرعي السابق لعملية البحث غابة تتكون من عدة أشجار بأولوية العمق، بحيث تمثل الحواف في E حوافاً لشجرة، وكما هو الحال في البحث بالاتساع أولاً، تعتمد خوارزمية البحث بالعمق على تلوين الرؤوس أثناء عملية الاستكشاف لتحديد حالتها؛ إذ يُعتبر كل رأس أبيض في البداية، ثم يتحوّل إلى اللون الرمادي عند اكتشافه، ويُصبغ بالأسود عند الانتهاء من فحص قائمة الجوار الخاصة به بالكامل. وتضمن هذه الآلية أن يُدرج كل رأس ضمن شجرة واحدة فقط من أشجار البحث، مما يجعل الأشجار الناتجة غير متقاطعة.

وبالإضافة إلى بناء غابة البحث، تقوم الخوارزمية أيضاً بوسم كل رأس بطابعين زمنيّين: الطابع الزمني الأول $u.d$ يُسجّل وقت اكتشافه لأول مرة (وتلوينه بالرمادي)، والطابع الزمني الثاني $u.f$ يُسجّل وقت انتهاء البحث من فحص قائمة جواره. هذه الطوابع الزمنية تُقدم معلومات مهمة حول بنية الرسم البياني، وهي مفيدة بشكل عام في فهم سلوك البحث بالعمق أولاً.

تقوم الإجرائية الخاصة بالبحث بالعمق (DFS) بتسجيل وقت اكتشاف كل رأس u في السمة $u.d$ ووقت الانتهاء منه في السمة $u.f$ ، ونظراً لوجود حدثيْن لكل رأس u: الاكتشاف والانتهاء، فإن العلاقة الزمنية تكون على النحو التالي: $u.d < u.f$ ، وبذلك، يكون الرأس u أبيض قبل الزمن $u.d$، ورمادياً بين الزمنين $u.d$ و $u.f$، ثم يتحوّل إلى الحالة السوداء بعد الزمن $u.f$.

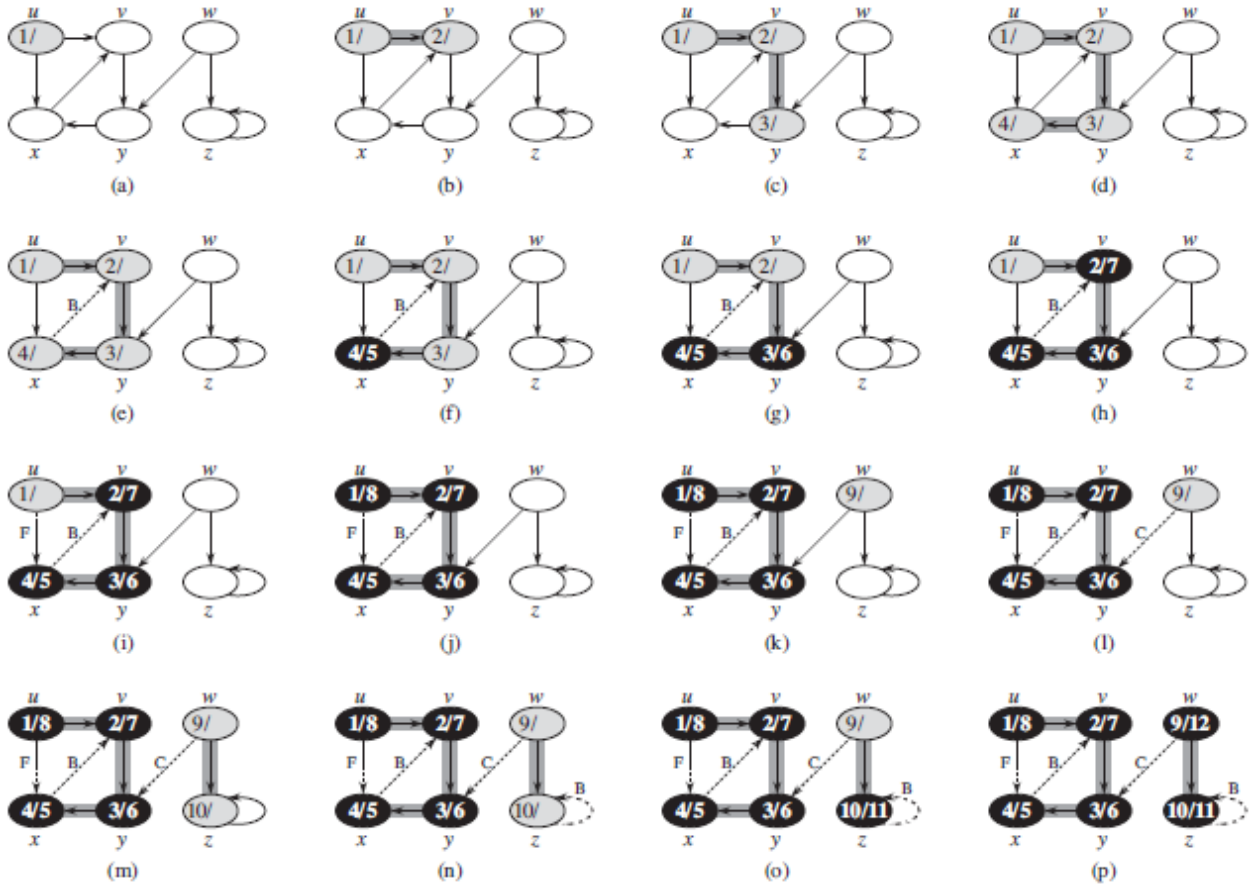
يمثل الكود الزائف (pseudocode) التالي الخوارزمية الأساسية للبحث بالعمق أولاً، حيث أن متغير الوقت (*time*) هو متغير عام نستخدمه للطابع الزمني، و G يمثل الرسم البياني، أما G.V فهو مجموعة الرؤوس داخل هذا الرسم البياني، وتمثل $G.Adj[u]$ قائمة الجوار للرأس u:

```
DFS (G)
1   for each vertex u ϵ G.V
2       u.color = WHITE
3       u.π = NIL
4   time = 0
5   for each vertex u ϵ G.V
6       if u.color = = WHITE
7           DFS-VISIT(G,u)

DFS-VISIT(G,u)
1   time = time + 1 // white vertex u has just been discovered
2   u.d = time
3   u.color = GRAY
4   for each v ϵ G. Adj[u] // explore edge (u,v)
5       if v.color = = WHITE
6           v. π = u
7           DFS-VISIT (G, v)
8   u.color = BLACK // blacken u; it is finished
9   time = time + 1
10  u.f = time
```

يوضح الشكل (4–1) كيفية تقدم عملية البحث بالعمق (DFS) والتي تعمل وفق الخطوات التالية:

في الأسطر (1–3) من الكود الزائف، يتم تلوين جميع الرؤوس باللون الأبيض، وتُهيّأ سماتها لتأخذ القيمة NIL. في السطر (4)، يُعاد ضبط العداد الزمني، ثم تقوم الأسطر (5–7) بفحص كل رأس في المجموعة V على التوالي، وعند العثور على رأس أبيض، يتم استدعاء الإجرائية الفرعية DFS-VISIT لزيارته. وكل مرة تُستدعى فيها DFS-VISIT للرأس u في السطر (7)، يُعتبر هذا الرأس جذراً لشجرة جديدة ضمن غابة البحث بالعمق. وعند انتهاء تنفيذ DFS، يكون كل رأس u قد حصل على طابع زمني للاكتشاف $u.d$ وطابع زمني للانتهاء $u.f$.



الشكل (4–1)– البحث بالعمق في خوارزمية DFS [13]

في كل استدعاء DFS-VISIT(G,u)، يكون الرأس u أبيض في البداية. يقوم السطر (1) بزيادة المتغيّر الزمني العام *time*، ثم يسجّل السطر (2) القيمة الجديدة كوقت اكتشاف $u.d$، ويقوم السطر (3) بتلوين الرأس u باللون الرمادي. بعد ذلك تفحص الأسطر (4 – 7) كل رأس مجاور لـ u وتزوره بشكل متكرر إذا كان أبيض. وكل رأس $v \in Adj[u]$ يُؤخذ في الاعتبار في السطر (4)، ونقول إن الحافة edge(u,v) تُستكشف بواسطة البحث بالعمق أولاً. وأخيراً، بعد استكشاف كل حافة خارجة من الرأس u، تقوم الأسطر (8 – 10) بتلوين u باللون الأسود، وتزيد الزمن، ثم تسجّل وقت الانتهاء $u.f$.

تجدر الإشارة إلى أن نتائج البحث بالعمق قد تختلف تبعاً لترتيب فحص الرؤوس في السطر (5) من DFS، وكذلك حسب ترتيب زيارة الجيران في السطر (4) من DFS-VISIT. ومع ذلك، فإن هذه الاختلافات في الترتيب لا تُسبب مشاكل عملية عادةً، إذ يمكن استخدام أي نتيجة للبحث بالعمق بشكل فعّال، وتكون النتائج متكافئة من حيث الجوهر.

أما بالنسبة للزمن التشغيلي لخوارزمية DFS، فإن الحلقات في الأسطر (1–3) و (5–7) تستغرق وقتاً من رتبة $\Theta(v)$، دون احتساب زمن تنفيذ الاستدعاءات لـDFS-VISIT، ونستخدم هنا التحليل التجميعي حيث يُستدعى إجراء DFS-VISIT مرة واحدة فقط لكل رأس $v \in V$، حيث يجب أن يكون الرأس u الذي يُستدعى عليه DFS-VISIT أبيض اللون، وأول خطوة تقوم بها الإجرائية هي تلوين u بالرمادي. وخلال تنفيذ DFS-VISIT للرأس v، تُنفّذ الحلقة في الأسطر (4–7) عدداً من المرات يساوي $|Adj[u]|$ مرة. وبناءً على ذلك فإن :

$$\sum_{v \in V} |Adj[v]| = \Theta(E)$$

والتكلفة الإجمالية لتنفيذ الأسطر (4 – 7) من DFS-VISIT هي $\Theta(E)$، وبالتالي فإن وقت تشغيل DFS هو $\Theta(V+E)$ [13]

الفصل الخامس

بروتوكولات التوجيه في الشبكات الحاسوبية

Routing protocols in networks

1.5. مقدمة لبروتوكولات التوجيه بالشبكات الحاسوبية

نمت شبكات الاتصال الحاسوبية بشكل كبير جداً في كل مكان بالعالم وتزايد نموها يوماً بعد يوم، ومع تزايد الحاجة لاتصال المستخدمين حول العالم بعضهم ببعض أو بقواعد البيانات لاستخلاص المعلومات أو نقل الملفات أو بث الفيديو أو المحادثات الصوتية أينما كانت في بقاع الأرض، حيث سهلت تكنولوجيا الاتصالات الحديثة عملية الوصول إلى مختلف مصادر المعلومات بشكل سريع وكبير بغض النظر عن مكان تواجدهم حول العالم. حيث يتم بناء شبكات الاتصال الحاسوبية باستخدام تقنيات مستفيدة من البنية التحتية للاتصالات ومن ضمنها بروتوكولات توجيه خاصة لنقل حزم البيانات عبر الشبكات الممتدة حول العالم.

2.5. أهمية بروتوكولات التوجيه

إن عملية التوجيه هي إرسال المعلومات من المصدر إلى الهدف باستخدام شبكات الاتصالات وتمرير هذه المعلومات عبر بعض الأجهزة الوسيطة. إن وظيفة بروتوكولات التوجيه هي منح هذه الأجهزة المعلومات اللازمة لإرسال حزم البيانات بشكل صحيح، فبدون هذه البروتوكولات فإن الأجهزة المختلفة التي تتكون منها الشبكة لا تستطيع المخاطبة فيما بينها وبالتالي لن تصل حزم البيانات إلى الهدف. فهذه البروتوكولات تحدد كيفية اتصال الأجهزة الوسيطة (الموجهات) مع بعضها البعض ونشر المعلومات التي تمكنها من تحديد الوجهة بين أي عقدتين في شبكة الاتصال الحاسوبية وذلك عن طريق خوارزميات محددة تدعى بخوارزميات التوجيه التي تحدد خيار الوجهة المطلوبة حيث كل موجه له معرفة مسبقة بالشبكات المتصلة به مباشرة، وتشارك بروتوكولات التوجيه هذه المعلومات أولاً عن طريق الجيران المباشرين ومن ثم عبر الشبكة. وبهذه الطريقة تزيد المعرفة بطوبولوجيا الشبكة التي يتصل بها الموجه.

وحتى تعمل بروتوكولات التوجيه بشكل صحيح، يجب أن تصف هذه الخوارزميات بعض الإجرائيات وهذه الإجرائيات هي:

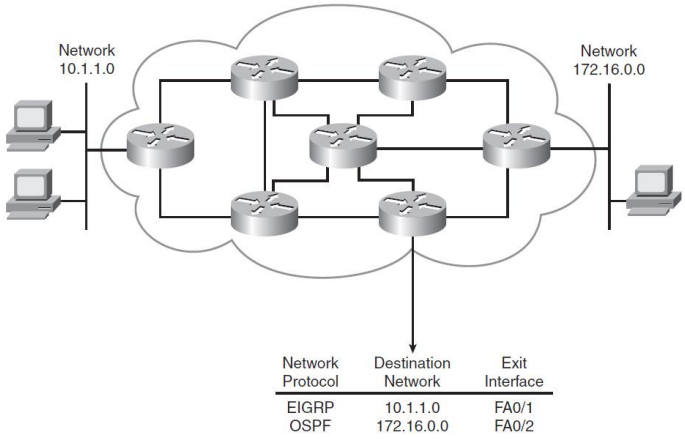
- إجرائية استقبال وإرسال المعلومات عبر الشبكة.
- إجرائية إيجاد أفضل مسار للهدف و وضع الوجهة في جدول التوجيه.
- إجرائية الاكتشاف والتفاعل و إخبار الأجهزة الأخرى عن التغييرات في طوبولوجيا الشبكة.

3.5. التوجيه Routing

يمكن أن تقوم الموجهات بإعادة توجيه الحزم ضمن شبكات الاتصالات عبر مسارات ثابتة أو ديناميكية وذلك بناءً على إعدادات الموجه.

هناك طريقتان لإعلام الموجه بكيفية إعادة توجيه الحزم إلى شبكات غير متصلة بشكل مباشر وهي:

- التوجيه الثابت (Static Routing): يتعلم الموجه المسارات عندما يقوم مسؤول الشبكة بإعداد المسار الثابت يدوياً ويجب عليه تحديث إدخاله بشكل يدوي عندما تتغير طوبولوجيا الشبكة، والمسارات الثابتة هي المسارات المعرفة من قبل المستخدم والتي تحدد المسار الذي تتخذه الحزم عند الانتقال بين المصدر والوجهة، وتتيح هذه المسارات المعرّفة من قبل مسؤول الشبكة التحكم الدقيق في طريقة توجيه الشبكة.
- التوجيه الديناميكي (Dynamic Routing): يتعلم الموجه المسارات آلياً بعد قيام مسؤول الشبكة بإعداد بروتوكول توجيه يساعد على تحديد هذه المسارات وعلى عكس الوضع مع المسارات الثابتة، بعد أن يقوم مسؤول الشبكة بتفعيل التوجيه الديناميكي، تقوم عملية التوجيه تلقائياً بتحديث المعلومات عن المسار وذلك كلما تم استلام معلومات طوبولوجيا جديدة (مثل إضافة موجه شبكة جار أو حذفه) حيث يتعلم الموجه المسارات إلى الوجهات البعيدة ويحافظ عليها عن طريق تبادل تحديثات معلومات التوجيه مع الموجهات الأخرى في الشبكة.
- يعتمد التوجيه الديناميكي على بروتوكول توجيه قابل للتعلم، حيث يحدد بروتوكول التوجيه القواعد التي يستخدمها الموجه عندما يتصل مع أجهزة التوجيه المجاورة لتحديد المسارات إلى الشبكات البعيدة ويحتفظ بمعلومات عنها في جداول التوجيه. ويبين الشكل التالي أن الموجه في الشبكة يمكن أن يكون لديه معرفة بالشبكات التي لا ترتبط مباشرة بأيّة بوابة على هذا الجهاز، حيث يجب إعداد هذه المسارات بشكل ثابت أو تعلمها عن طريق بروتوكولات التوجيه [14].



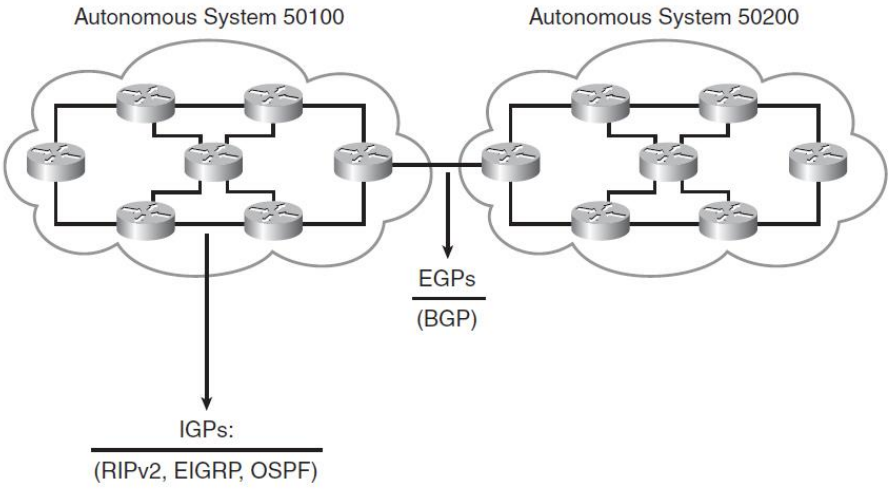
الشكل (5-1)- توجيه البروتوكولات [14]

4.5. أنواع بروتوكولات التوجيه

يوجد نوعان لبروتوكولات التوجيه:

- بروتوكولات البوابة الداخلية IGP (Interior gateway protocol): تتبادل بروتوكولات التوجيه معلومات التوجيه داخل نظام مستقل واحد. والبروتوكولان EIGRP و OSPF هما مثالان عن هذه البروتوكولات.
- بروتوكولات البوابة الخارجية EGP (Exterior Gateway Protocol): وتستخدم هذه البروتوكولات في التوجيه بين الأنظمة المستقلة، وبروتوكول البوابة الحدودية BGP (Border Gateway Protocol) هو الاختيار الحالي بالشبكات في يومنا الحاضر عند استعمال هذا الخيار.

والشكل التالي يوضح بنية كلا البروتوكولين السابقين:



الشكل (5-2)- الفرق بين التوجيه الداخلي والخارجي [14]

يمكن تصنيف معظم توجيهات IGP ضمن النظام المستقل الواحد على أنها مطابقة لإحدى الخوارزميات التالية:

- شعاع المسافة (Distance vector): يحدد الاتجاه (Vector) والمسافة (مثل القفزات) لأي وصلة في الشبكة.
- حالة الوصلة (Link-state): يستخدم خوارزمية المسار الأقصر SPF (Shortest Path First)، حيث ينشئ فكرة عامة للطوبولوجيا الدقيقة للشبكة بأكملها، أو على الأقل، القسم الذي يوجد فيه الموجه.
- شعاع المسافة المحسّن (Advanced distance vector): وهو يجمع بين جوانب الخوارزميتين حالة الوصلة وشعاع المسافة. ويشار إليه أحياناً باسم بروتوكول التوجيه الهجين.

توجد بشكل عام وفق طوبولوجيا الشبكة طرق متعددة للوصول إلى الوجهة، وعندما تقوم خوارزمية بروتوكول التوجيه بتحديث جدول التوجيه، فإن الهدف الأساسي للخوارزمية هو تحديد أفضل مسار ليتم تضمينه داخل الجدول، فمثلاً يستخدم بروتوكول شعاع المسافة مقياس توجيه مختلف لتحديد أفضل مسار، حيث تقوم الخوارزمية بإنشاء رقم يسمى قيمة المقياس لكل مسار عبر الشبكة، ويمكن حساب المقاييس استناداً إلى خاصية واحدة للمسار، ومن أجل المقاييس الأكثر تعقيداً فيتم حسابها من خلال الجمع بين عدة خصائص للمسار.

المقاييس التي تستخدمها بروتوكولات التوجيه الأكثر شيوعاً هي كما يلي:

عدد القفزات: عدد المرات التي تمر فيها الحزمة عبر بوابة جهاز التوجيه.

عرض الحزمة: سعة البيانات لوصلة، فعلى سبيل المثال، عادة ما تكون وصلة إيثرنت 10 ميغابت في الثانية أفضل من خط مؤجر 64 كيلوبت في الثانية.

التأخير: الوقت المطلوب لنقل الحزمة من المصدر إلى الوجهة .

الحمل: مقدار النشاط على موارد الشبكة، مثل جهاز التوجيه أو الوصلة .

الوثوقية: تشير عادة إلى معدل الخطأ في البتات لكل وصلة شبكة .

التكلفة: قيمة قابلة للإعداد على أجهزة التوجيه وتستند بشكل افتراضي إلى عرض الحزمة للبوابة، حيث تلعب التكلفة دوراً أساسياً في اختيار أفضل مسار.

5.5. خوارزميات التوجيه

يتم تحديد كافة بروتوكولات التوجيه على أساس خوارزميتها، هذه الخوارزمية يجب أن تصف بعض الإجراءات من أجل جعل بروتوكولات التوجيه تعمل بشكل صحيح، وهذه الإجراءات هي:

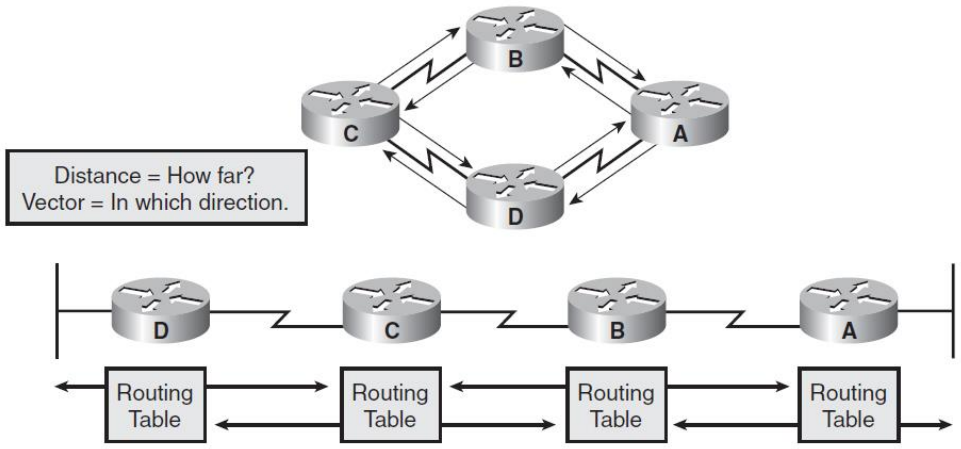
- استقبال وإرسال المعلومات حول الشبكة.
- إيجاد أفضل مسار للوصول إلى الوجهة وتثبيت المسار في جدول التوجيه.
- وأخيرا، الكشف عن رد فعل الأجهزة وإبلاغ الأجهزة الأخرى حول التغييرات في الشبكة.

ومن أشهر خوارزميات التوجيه الرئيسية الموجودة، هي خوارزمية بيلمان–فورد (Bellman-Ford) وخوارزمية ديكسترا (Dijkstra) واعتماداً على الخوارزمية المستخدمة تصنف بروتوكولات التوجيه إلى شعاع المسافة (Distance Vector) أو حالة الوصلة (Link State) على التوالي، أما الفرق الأساسي بينهما فهو كيفية تنفيذ الإجراءات المذكورة أعلاه، وفيما يلي عرض تفصيلي للاختلافات بين هذه الخوارزميات.

5. 5. 1. خوارزمية شعاع المسافة (Distance Vector Algorithm)

تقوم خوارزميات التوجيه المستندة إلى شعاع المسافة (المعروفة أيضا باسم خوارزمية بلمان–فورد) بتمرير نسخ دورية لجدول التوجيه من موجه إلى موجه، وتصل التحديثات المنتظمة بين أجهزة التوجيه عند تغير الطوبولوجيا حيث يتلقى

كل موجه جدول التوجيه من جاره المباشر. فعلى سبيل المثال، في الشكل التالي، يتلقى الموجه B معلومات من الموجه A، ويضيف الموجه B مقياس شعاع المسافة (مثل عدد القفزات)، مما يزيد من شعاع المسافة، ثم يمرر جدول التوجيه إلى جار آخر وهو الموجه C. تحدث هذه العملية خطوة بخطوة في جميع الاتجاهات بين أجهزة التوجيه المتجاورة، (وهذا ما يعرف أيضاً باسم التوجيه عن طريق الترويج أو الإشاعة).



الشكل (5-3)- خوارزمية شعاع المسافة [14]

وبهذه الطريقة، تراكم الخوارزمية مسافات الشبكة بحيث يمكنها الحفاظ على قاعدة بيانات عن معلومات طوبولوجيا الشبكة بينما لا تسمح هذه الخوارزمية لجهاز التوجيه بمعرفة الطوبولوجيا الدقيقة للشبكة.

1.1.5.5. اكتشاف الطريق والاختيار

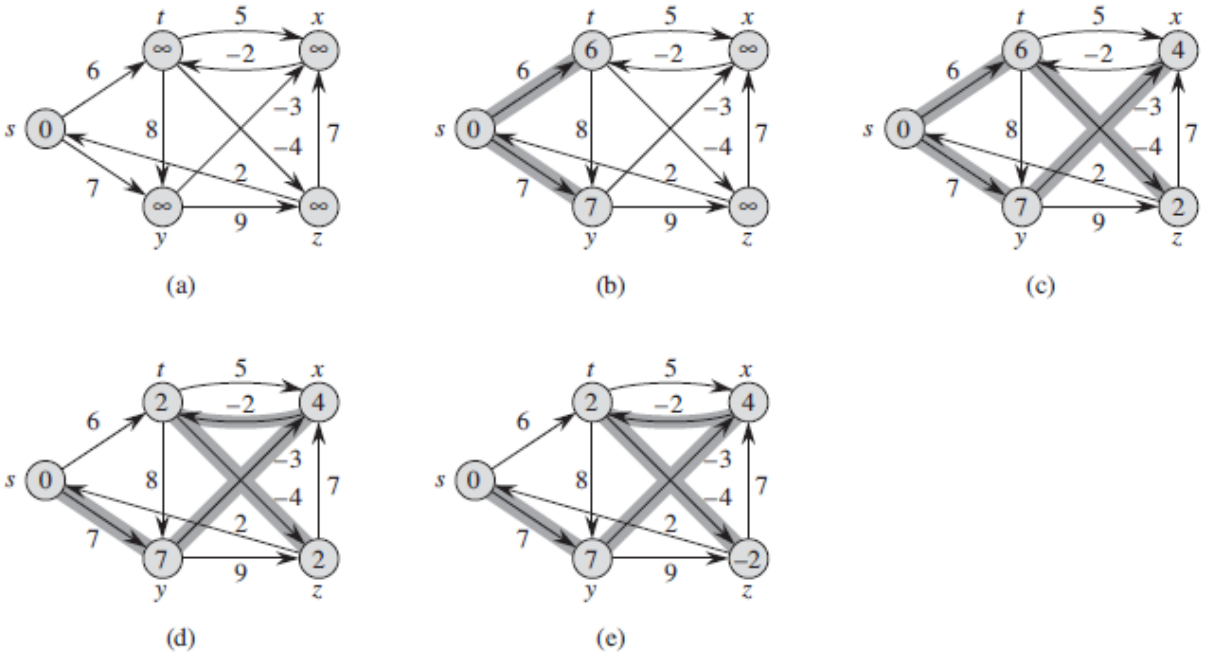
- تعتمد خوارزمية التوجيه بشعاع المسافة على قيام كل موجه ببناء جدول يتضمن أفضل مسافة معروفة تفصل بينه وبين كل هدف ممكن والمسار الذي عليه استخدامه للوصول إلى ذلك الهدف، ويتم تحديث هذه الجداول بصورة مستمرة عن طريق تبادل المعلومات مع الموجهات المتجاورة، ويطلق على هذه الخوارزمية عدة أسماء الأكثر شيوعاً هي خوارزمية Bellman-Ford الموزعة.
- يقوم كل موجه ببناء وتحديث جدول مبني حسب عدد الموجهات الموجودة في الشبكة بحيث يحوي مدخلاً واحداً لكل موجه، ويتألف هذا المدخل من جزأين أساسيين:

- خط الخرج المفضل استخدامه للوصول إلى هذا الهدف
- قيمة تقديرية للوقت أو المسافة بين الموجه وذلك الهدف

- يمكن أن يكون هذا المقياس المستخدم عبارة عن عدد القفزات، مقدار التأخير الزمني (ms) أو العدد الإجمالي للحزم المجمعة بانتظار تحديد دورها عبر المسار وبشكل عام يتم الاعتماد على التأخير الزمني اللازم للوصول إلى الموجه الجار.
- تحل خوارزمية بيلمان–فورد مسألة أقصر المسارات أحادية المصدر وذلك في الحالة العامة التي قد تتضمّن أوزاناً سالبة للحواف. بافتراض رسم بياني موجه $G= (V,E)$ مع المصدر s ودالة الوزن $w: E \longrightarrow R$، تُرجع خوارزمية بيلمان–فورد قيمة منطقية تُشير ما إذا كانت هناك دورة سالبة الوزن يمكن الوصول إليها من المصدر. في حال وجود مثل هذه الدورة، تُعلن الخوارزمية إلى عدم وجود حل. أما في حال عدم وجودها، فتُنتج الخوارزمية أقصر المسارات وأوزانها.
- تقوم الخوارزمية بمعالجة الأضلاع تدريجياً، بحيث تُحدّث تقدير المسافة v.d لأقصر مسار من المصدر s إلى كل رأس $v \in V$ حتى تصل إلى الوزن الفعلي لأقصر مسار $\delta(s,v)$ ، تُرجع الخوارزمية القيمة TRUE إذا وفقط إذا لم تكن هناك دورات ذات أوزان سالبة يمكن الوصول إليها من المصدر.

```
BELLMAN-FORD (G, w, s)
1  INITIALIZE-SINGLE-SOURCE (G, s)
2  for i = 1 to |G.V| _ 1
3      for each edge (u,v) ϵ G.E
4          RELAX (u,v,w)
5  for each edge (u,v) ϵ G.E
6      if v.d > u.d + w(u,v)
7                  return FALSE
8  return TRUE
```

- يوضح الشكل (5–4) تنفيذ خوارزمية بيلمان–فورد على رسم بياني ذي خمسة رؤوس. بعد تهيئة قيم d و π لجميع الرؤوس في السطر الأول، تُجري الخوارزمية $|V|-1$ مروراً على حواف الرسم البياني. كل مرور هو تكرار واحد لحلقة for للأسطر (2–4)، ويتكون من معالجة كل حافة من حواف الرسم البياني مرة واحدة. توضح الأشكال (5–4): (b) – (e) حالة الخوارزمية بعد كل من التمريرات الأربع على الحواف. بعد الانتهاء من $|V|-1$ تمريرة، تقوم الأسطر (5–8) بالتحقق من وجود دورة سالبة الوزن، وتُرجع القيمة المنطقية المناسبة.
- تعمل خوارزمية بيلمان–فورد في الزمن $O(VE)$، حيث تستغرق عملية التهيئة في السطر الأول زمناً من رتبة $\Theta(V)$، ويستغرق كل من التمريرات $|V|-1$ على الحواف في الأسطر من (2 – 4) زمناً من رتبة $\Theta(E)$ ، كما أن حلقة التكرار للأسطر (5 – 7) تستغرق زمناً من رتبة $O(E)$.
- لإثبات صحة خوارزمية بيلمان–فورد، نبدأ بإثبات أنه في حال عدم وجود دورات ذات أوزان سالبة، تحسب الخوارزمية أوزاناً صحيحة لأقصر مسار لجميع الرؤوس التي يمكن الوصول إليها من المصدر.



الشكل (5–4)– تنفيذ خوارزمية بيلمان–فورد [13]

- يتم تنفيذ خوارزمية بيلمان–فورد حيث المصدر هو الرأس s وتظهر قيم d داخل الرؤوس، بينما تشير الحواف المظللة إلى القيم السابقة: فإذا كانت الحافة (u,v) edge مظللة، فإن $v.\pi = u$ وفي هذا المثال تحديداً، يتم في كل دورة معالجة الأضلاع بالترتيب التالي:
- $(t,x),(t,y),(t,z),(x,t),(y,x),(y,z),(z,x),(z,s),(s,t),(s,y)$ (a) الوضع قبل أول عملية تمرير فوق الحواف. (b) – (e) الوضع بعد كل عملية تمرير متتالية فوق الحواف. القيمتان d وπ في الجزء (e) هما القيمتان النهائيتان. وتُرجع خوارزمية بيلمان–فورد القيمة TRUE في هذا المثال.[13]

2.1.5.5. بروتوكولات شعاع المسافة (Distance Vector Protocols)

يعتبر البروتوكول RIP من أهم البروتوكولات التي تعمل وفق خوارزمية شعاع المسافة وفيما يلي شرح خصائصه الرئيسية:

1.2.1.5.5. البروتوكول RIP (Routing Information Protocol)

وهو بروتوكول شعاع مسافة حقيقي، يقوم بإرسال كامل جدول التوجيه لكل البوابات الفعالة كل 30 ثانية، ويعتمد على عدد الموجهات لتحديد أفضل مسار للشبكات البعيدة، ولكن له عدد أعظمي افتراضي من القفزات المسموحة ألا وهو 15 قفزة، ونتيجة لذلك فهو يعمل بشكل جيد في الشبكات الصغيرة وتقل فعاليته في الشبكات الكبيرة التي تتضمن وصلات WAN (Wide Area Network) بطيئة أو عدد كبير من الموجهات.

يستخدم RIP التوجيه classful فقط، مما يعني أنه يجب على جميع الأجهزة في الشبكة استخدام قناع شبكة واحد وذلك بسبب أن RIP لا يرسل التحديثات مع معلومات قناع الشبكة.

1.1.2.1.5.5. عدادات البروتوكول RIP

يستخدم بروتوكول RIP أربعة أنواع مختلفة من العدادات الزمنية لتنظيم الأداء وهي:

- Route Update Timer: وهو يحدد المدة الزمنية بين فترات إرسال التحديثات (30 ثانية) لكل الموجهات المجاورة.

- Route Invalid Timer: وهو يحدد الوقت الذي بانقضائه يعتبر مسار ما غير ممكن الوصول إليه (180 ثانية) وذلك إذا لم يتم استلام أي تحديثات عنه خلال هذا الزمن، وعندما يحدث ذلك لأي مسار يرسل الموجه معلومات لكل جيرانه بأن المسار لا يمكن الوصول إليه.
- Hold-down Timer: وهو يحدد مقدار الوقت الذي يتم خلاله إهمال البيانات الموجهة. وتدخل الموجهات فترة توقف عندما يتم استقبال حزمة تحديثات تبين بأن المسار غير ممكن الوصول إليه، وهذه الفترة تستمر إلى أن يتم استقبال تحديثات بتكلفة مسار أقل أو إلى أن يتم انتهاء فترة العداد (الافتراضي 180 ثانية).
- Route flush timer: وهو يحدد الوقت بين اعتبار مسار ما غير صالح وبين إزالته من جدول التوجيه (الزمن الافتراضي 240 ثانية)، وقبل إزالة المسار من جدول التوجيه ينبّه الموجه جيرانه بإنهاء ذلك المسار [14].

2.1.2.1.5.5. الطلبات (Requests)

يُستخدم الطلب (Request) لطلب استجابة تحتوي على كل أو جزء من جدول توجيه المضيف .عادةً، تُرسل الطلبات كبث، من منفذ مصدر UDP (User Datagram Protocol) رقمه 520.

يُعالج الطلب مدخلاً تلو الآخر، وإذا لم تكن هناك مدخلات، فلن تُعطى أي استجابة، وهناك حالة خاصة واحدة، إذا كان هناك مُدخل واحد فقط في الطلب، مع مُعرّف عائلة عنوان 0 (أي غير مُحدد)، ومقياس لانهائي (أي 16 للتطبيقات الحالية)، فهذا طلب لإرسال جدول التوجيه بالكامل . في هذه الحالة، يتم استدعاء عملية الإخراج لإرسال جدول التوجيه إلى المنفذ الطالب .

باستثناء هذه الحالة الخاصة، تكون المعالجة بسيطة للغاية، حيث ينتقل إلى قائمة المُدخلات في الطلب واحداً تلو الآخر، ويبحث عن الوجهة في قاعدة بيانات توجيه المُضيف. إذا كان هناك مسار، فيضع مقياس هذا المسار في حقل المقياس في مخطط البيانات، وإذا لم يكن هناك مسار إلى الوجهة المُحددة، فيضع لانهائي (أي 16) في حقل المقياس، وبمجرد ملء جميع المُدخلات، يضبط الأمر على "استجابة" ويرسل مخطط البيانات مرة أخرى إلى المنفذ الذي جاء منه .

3.1.2.1.5.5. الاستجابة

يمكن استلام الاستجابات لأسباب مختلفة :

- استجابة لاستعلام محدد
- تحديثات منتظمة
- تحديثات مُفعّلة نتيجة تغيير في المقياس

يتم التحقق من صحة الاستجابة حيث يتم تجاهل الاستجابة إذا لم تكن من المنفذ 520 والتحقق من عنوان مصدر IP (Internet Protocol) لمعرفة ما إذا كانت حزمة البيانات من جهاز جار صالح، وأيضاً أن يكون مصدر حزمة البيانات على شبكة متصلة مباشرةً.

4.1.2.1.5.5. أهم ميزات البروتوكول

يتم إنشاء جدول توجيه لكل مضيف يُطبق بروتوكول RIP. يحتوي هذا الجدول على مُدخل واحد لكل وجهة يُمكن الوصول إليها عبر النظام الموصوف في بروتوكول RIP.

يحتوي كل مُدخل على الأقل على المعلومات التالية:

- عنوان IP للوجهة.
- مقياس يُمثل التكلفة الإجمالية لتوصيل مُخطط بيانات من المُضيف إلى تلك الوجهة. هذا المقياس هو مجموع التكاليف المُرتبطة بالشبكات التي سيتم اجتيازها للوصول إلى الوجهة.
- عنوان IP للبوابة التالية على طول المسار المُؤدي إلى الوجهة.
- علامة تُشير إلى أن معلومات المسار قد تغيّرت مُؤخراً. يُشار إلى هذه العلامة باسم "علامة تغيير المسار".
- مُؤقتات مُختلفة مُرتبطة بالمسار.
- يُنشئ المُضيف مُدخلات الشبكات المُتصلة مُباشرةً، باستخدام المعلومات التي يتم تجميعها، في تطبيقات RIP الحالية، يُستخدم الرقم (1) دائماً للتكلفة وفي هذه الحالة، يُختزل مقياس RIP إلى عدد قفزات بسيط حيث يمكن

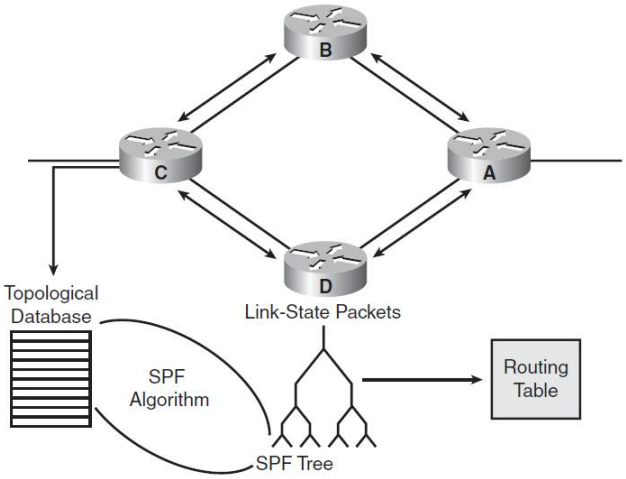
استخدام مقاييس أكثر تعقيداً عندما يكون المطلوب إظهار تفضيل بعض الشبكات على غيرها، على سبيل المثال بسبب اختلافات في عرض النطاق الترددي أو الوثوقية.

5.1.2.1.5.5. قيود البروتوكول RIP

- يقتصر البروتوكول على الشبكات التي يتضمن أطول مسار لها 15 قفزة.
- يعتمد البروتوكول على "العد إلى ما لا نهاية" لحل بعض المواقف غير العادية، فإذا كانت الشبكة تضم عدة مئات من الشبكات، وتم تشكيل حلقة توجيه تشملها جميعاً، فسيتطلب حل الحلقة إما وقتاً طويلاً (إذا كان تواتر تحديثات التوجيه محدوداً) أو نطاقاً ترددياً كبيراً (إذا تم إرسال التحديثات كلما تم اكتشاف تغييرات) وستستهلك هذه الحلقة قدراً كبيراً من عرض حزمة الشبكة قبل تصحيحها.
- من مساوئ هذا البروتوكول كونه يستخدم "محددات" ثابتة لمقارنة المسارات البديلة وذلك غير مناسب للحالات التي تتطلب اختيار المسارات بناءً على محددات آنية مثل التأخير، أو الوثوقية، أو الحمل.

2.5.5. التوجيه حسب حالة الوصلة

بالإضافة إلى التوجيه بشعاع المسافة فإن الطريقة الثانية المستخدمة في التوجيه هي التوجيه حسب حالة الوصلة. وتبني بروتوكولات حالة الوصلة جداول التوجيه استناداً إلى قاعدة بيانات طوبولوجيا الشبكة، فيما يتم إنشاء قاعدة البيانات من حزم حالة الوصلة التي يتم تمريرها بين جميع أجهزة التوجيه لوصف حالة الشبكة. وتستخدم خوارزمية (أقصر مسار أولاً) قاعدة البيانات لبناء جدول التوجيه، وذلك كما هو مبين في الشكل (5-5) :



الشكل (5-5)- التوجيه حسب حالة الوصلة [14]

وتعرف خوارزمية حالة الوصلة أيضاً باسم خوارزمية المسار الأقصر أولاً (SPF) حيث تحتفظ بقاعدة بيانات معقدة من معلومات لطوبولوجيا الشبكة، في حين أن خوارزمية شعاع المسافة ليس لديها معلومات محددة عن طوبولوجيا الشبكات البعيدة ولا توجد لديها معرفة بالموجهات البعيدة، فإن خوارزمية التوجيه لحالة الوصلة تحافظ على معرفة كاملة بالموجهات البعيدة وكيفية ربطها مع بعضها البعض. لأنها تستخدم إعلانات حالة الوصلة (LSA)، وخوارزمية وشجرة نتائج المسار الأقصر أولاً، بالإضافة إلى جدول توجيه المسارات والبوابات لكل شبكة [14].

5. 5. 3. خوارزمية ديكسترا (Dijkstra):

تُعد خوارزمية ديكسترا من أكثر الخوارزميات شيوعاً ضمن منهجية التوجيه المعتمد على الوصلة (Link-State Routing)، حيث تعتمد على وسم كل عقدة في الشبكة بمسافة تقديرية تفصلها عن عقدة المصدر، إلى جانب أفضل مسار معروف للوصول إليها. في بداية التنفيذ، لا تكون هناك أي معلومات متوفرة عن المسارات، لذا تُوسم جميع العقد بقيمة "لانهاية" للدلالة على عدم المعرفة. ومع بدء الخوارزمية واكتشاف المسارات، تبدأ هذه الوسوم بالتغيّر تدريجياً لتعكس المسارات الأكثر كفاءة. وعندما يُكتشف أن مساراً معيناً يُمثّل أقصر طريق محتمل من المصدر إلى عقدة ما، يُعتمد هذا المسار بشكل نهائي ولا يُعاد تغييره لاحقاً.

تُعالج خوارزمية ديكسترا مسألة إيجاد أقصر المسارات انطلاقاً من مصدر واحد في رسم بياني موجّه ذي أوزان غير سالبة، حيث نفترض أن $w(u,v) \geq 0$ لكل حافة eadge $(u,v) \epsilon E$ وبفضل بنيتها الفعّالة، فإن زمن تنفيذها غالباً ما يكون أقل من زمن تنفيذ خوارزمية بيلمان–فورد.

تحافظ خوارزمية ديكسترا على مجموعة S من الرؤوس التي تم تحديد أوزان أقصر مسارات نهائية لها انطلاقاً من المصدر s.

تختار الخوارزمية بشكل متكرر الرأس u ϵ V- S الذي يمتلك أدنى تقدير للمسافة، ثم تضيفه إلى المجموعة S، وتتم معالجة جميع الحواف المرتبطة به. وفي التنفيذ العملي، نستخدم قائمة انتظار ذات أولوية دنيا Q تحتوي على الرؤوس التي يتم ترتيبها وفقاً لقيم d الخاصة بها والتي تمثل التقديرات الحالية لأوزان أقصر المسارات.

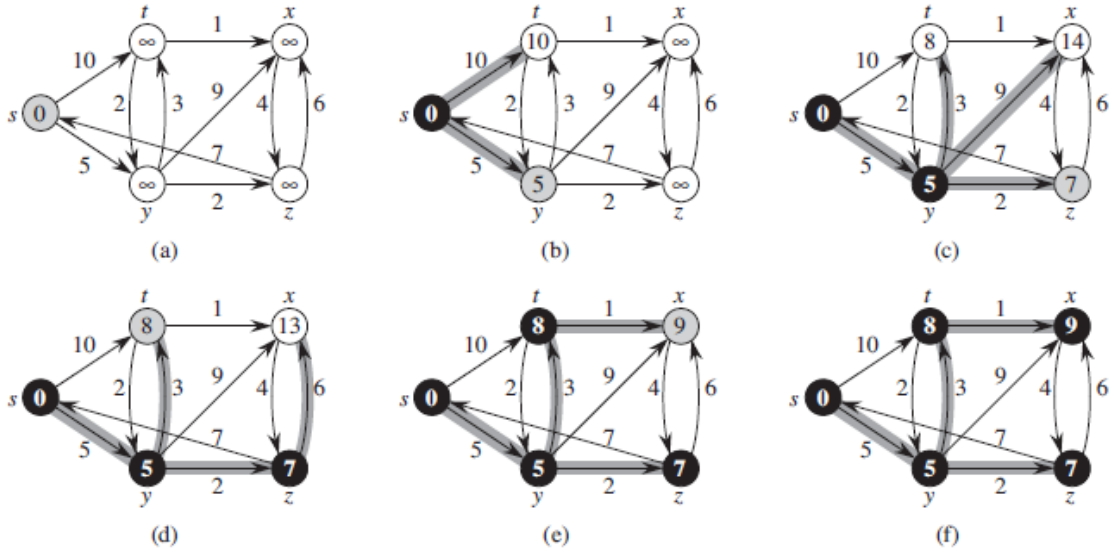
```
DIJKSTRA (G,w,s)
1  INITIALIZE-SINGLE-SOURCE (G, s)
2  S = Ø
3  Q = G.V
4  while Q ≠ Ø
5    u = EXTRACT-MIN (Q)
6    S = S Y {u}
7    for each vertex v ϵ G.Adj[u]
8          RELAX (u, v, w)
```

وفيما يلي شرح لهذه الخوارزمية:

تقوم خوارزمية ديكسترا بمعالجة الحواف كما هو موضح في الشكل (5–6) حيث يُهيئ السطر الأول قيم d و π وفق الطريقة المعتادة، بينما يُهيئ السطر الثاني المجموعة S لتكون فارغة. وتُحافظ الخوارزمية على شرط ثابت يتمثل في أن $Q = V- S$ عند بداية كل دورة من الحلقة التكرارية في الأسطر (4–8). في السطر الثالث، تُهيّأ قائمة الانتظار ذات الأولوية الدنيا Q تحتوي على جميع رؤوس المجموعة V؛ وبما أن $S=Ø$ في تلك اللحظة، فإن الشرط الثابت ($Q = V- S$) يتحقق بعد السطر الثالث.

في كل دورة من الحلقة التكرارية (الأسطر 4–8)، يقوم السطر الخامس باختيار الرأس u من المجموعة $Q = V - S$ والذي يمتلك أقل تقدير للمسافة، ثم يُضاف هذا الرأس إلى المجموعة S في السطر السادس، ما يُبقي على تحقق الشرط الثابت. (في الدورة الأولى خلال هذه الحلقة، يكون $u = s$ أي رأس المصدر) وبالتالي يكون الرأس u و الأقرب إلى المصدر من بين جميع الرؤوس غير المعالجة. بعد ذلك، تقوم الأسطر (7–8) بتحسين تقديرات المسارات لكل رأس v مجاور لـ u، وذلك من خلال تحديث القيمة $v.d$ والسلف $v.\pi$ إذا أمكن الوصول إلى v عبر u بطريقة أفضل من التقدير السابق. ومن الجدير بالملاحظة أن الخوارزمية لا تُدرج أي رؤوس جديدة في القائمة Q بعد السطر الثالث، وأن كل رأس يُستخرج من Q ويُضاف إلى S مرة واحدة فقط، مما يعني أن الحلقة التكرارية (الأسطر 4–8) تُنفّذ عدداً من المرات يساوي $|V|$.

و بما أن خوارزمية ديكسترا تختار دائماً الرأس "الأقرب" أو "الأخف وزناً" من بين الرؤوس في $V- S$ لإضافته إلى المجموعة S، فإنها تُصنّف ضمن الخوارزميات الجشعة (Greedy Algorithms). وعلى الرغم من أن هذا النوع من الاستراتيجيات لا يضمن دائماً الوصول إلى الحل الأمثل في جميع الحالات، إلا أن النظرية التالية ونتيجتها التبعية تُثبتان أن خوارزمية ديكسترا تُنتج بالفعل أقصر المسارات. ويتمثل جوهر الإثبات في أن كل مرة يُضاف فيها رأس u إلى المجموعة S، يكون التقدير $u.d$ مساوياً للقيمة الفعلية لأقصر مسار من المصدر، أي $u.d = \delta(s, u)$



الشكل (5–6)– تنفيذ خوارزمية ديكسترا [13]

يوضح الشكل (5–6) تنفيذ خوارزمية ديكسترا حيث المصدر s هو الرأس الأيسر وتظهر تقديرات أقصر مسار داخل الرؤوس، بينما تشير الحواف المظللة من كل شكل إلى قيم سابقة.

تشير الرؤوس السوداء إلى المجموعة S، في حين تمثّل الرؤوس البيضاء العناصر الموجودة في قائمة الانتظار ذات الأولوية الدنيا $Q = V - S$ ، في الجزء (a)، يُعرض الوضع قبل أول دورة من الحلقة التكرارية في الأسطر (4–8)، حيث يُختار الرأس المظلّل باعتباره الرأس u في السطر (5) نظراً لامتلاكه أدنى قيمة d. أما الأجزاء (b) إلى (f)، فتمثّل الحالة بعد كل دورة متتالية من الحلقة التكرارية، حيث يُختار الرأس المظلّل في كل جزء ليكون الرأس u في السطر (5) من الدورة التالية. وتُظهر القيم d وعلاقات السلف في الجزء (f) النتائج النهائية التي توصّلت إليها الخوارزمية [13] .

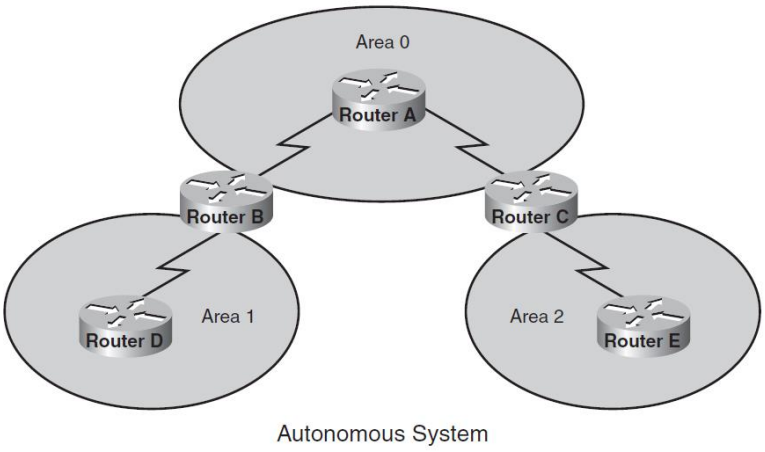
يتم تصنيف البروتوكول أقصر مسار مفتوح أولاً (OSPF) على أنه من بروتوكولات توجيه حالة الوصلة حيث تقوم هذه البروتوكولات بجمع معلومات التوجيه من جميع أجهزة التوجيه الأخرى في الشبكة أو داخل منطقة محددة من الشبكة، وبعد جمع كل المعلومات، يحسب كل جهاز توجيه وبشكل مستقل عن أجهزة التوجيه الأخرى أفضل مساراتها لجميع الوجهات في الشبكة. ولأن كل جهاز توجيه يحافظ على منظور خاص به عن الشبكة، فمن المحتمل نشر معلومات غير صحيحة مقدمة من قبل أي جهاز توجيه مجاور.

لقد تم تصميم بروتوكولات توجيه حالة الوصلة للتغلب على القيود المفروضة في بروتوكولات التوجيه لخوارزمية شعاع المسافة، حيث تستجيب بروتوكولات توجيه حالة الوصلة بسرعة لتغييرات الشبكة ولا ترسل التحديثات إلا عند حدوث تغيير في الشبكة ومن ثم ترسل تحديثات دورية (تعرف بتنبيهات حالة الوصلة) على فترات زمنية طويلة، كل (30) دقيقة، وتحدد آلية (hello) إمكانية الوصول للجيران. وعندما يحدث فشل في الشبكة، أي عند عدم القدرة على الوصول للموجهات الجيران، فإن بروتوكولات حالة الوصلة تحدث فيض (LSA) باستخدام عنوان (multicast) في المنطقة، ثم يأخذ كل جهاز توجيه نسخة من (LSA)، ويحدّث قاعدة بيانات حالة الوصلة، ومن ثم يعيد توجيه (LSA) لجميع الأجهزة المجاورة. وتتسبب (LSA) في كل جهاز توجيه داخل المنطقة بإعادة حساب الوجهات، ولأن (LSA) تحتاج إلى أن تعمم في جميع أنحاء المنطقة في الشبكة، وبالتالي تحتاج جميع أجهزة التوجيه داخل تلك المنطقة إلى إعادة

حساب جداول التوجيه الخاصة بها، وفي هذه الحالة يجب تقليل عدد أجهزة التوجيه المستخدمة لحالة الوصلة التي يمكن أن تكون في منطقة واحدة. ومن جهة أخرى تتشابه الوصلة مع المنفذ على جهاز التوجيه، فحالة الوصلة هي وصف لهذا المنفذ وعلاقته مع أجهزة التوجيه المجاورة له، ويتضمن وصف المنفذ، على سبيل المثال، عنوان IP للمنفذ والقناع ونوع الشبكة الموصولة به والموجهات المتصلة بهذه الشبكة وما إلى ذلك. وتشكل مجموعة حالات الوصلات قاعدة بيانات حالة الوصلة أو قاعدة بيانات طوبولوجيا الشبكة حيث تستخدم لحساب أفضل المسارات إلى الوجهة المقصودة ضمن الشبكة. وتجد موجهات حالة الوصلة أفضل المسارات إلى الوجهة عن طريق تطبيق خوارزمية ديكسترا لبناء شجرة المسار الأقصر أولاً (SPF). ثم يتم اختيار أفضل المسارات من هذه الشجرة ووضعها ضمن جدول التوجيه.

وعندما تصبح الشبكات أكبر حجماً، تصبح بروتوكولات توجيه حالة الوصلة أكثر جذباً وذلك للأسباب التالية:

- تقوم بروتوكولات حالة الوصلة دائما بإرسال التحديثات عند التغييرات الحاصلة في طوبولوجيا الشبكة.
- تكون التحديثات الدورية أقل تكراراً من تلك الموجودة في بروتوكولات شعاع المسافة.
- يمكن تقسيم الشبكات التي تعمل بواسطة بروتوكولات توجيه حالة الوصلة إلى مناطق هرمية ، مما يقلل من مجال تغيرات الوجهة.
- تمتاز الشبكات التي تستخدم بروتوكولات توجيه حالة الوصلة بدعمها العنونة بدون تصنيف (Classless Addressing).



الشكل (5-7)- البنية الهرمية لبروتوكولات حالة الوصلة [14]

ويوضح الشكل (5–7) تصميم OSPF نموذجي وبسيط حيث تحتوي هرمية بروتوكولات هذه الخوارزمية على عنصرين أساسيين:

- المنطقة (Area): هي مجموعة من الشبكات، والمناطق هي تقسيمات منطقية للنظام المستقل (AS).
- النظام المستقل (AS: Autonomous system): يتكون من مجموعة من الشبكات تحت إدارة مشتركة واحدة، حيث تشترك في استراتيجية توجيه واحدة، وتسمى AS أحياناً بالمجال، ويمكن تقسيمها منطقياً إلى عدة مناطق.

ويجب تحديد المنطقة الأساسية المجاورة ضمن كل منطقة AS، وترتبط جميع المناطق الأخرى غير الأساسية بالمنطقة الأساسية. والمنطقة الأساسية هي منطقة الانتقال، لأن جميع المناطق الأخرى تتصل من خلالها. أما بالنسبة لـ OSPF، فيمكن إعداد المناطق غير الأساسية كمنطقة فرعية، وذلك للمساعدة في تقليل قاعدة بيانات حالة الوصلة وحجم جدول التوجيه. [14]

4.5.5. بروتوكولات حالة الوصلة (Link State Protocols)

على عكس بروتوكولات شعاع المسافة فإن بروتوكولات حالة الوصلة تطور وتحافظ على المعرفة الكاملة لأجهزة التوجيه في الشبكة وكيفية ترابطها، ويتحقق ذلك من خلال تبادل حزم حالة الوصلة (LSA: Link State Advertisement) مع الموجهات الأخرى في الشبكة.

كل جهاز توجيه يتبادل LSAs فإنه يبني قاعدة بيانات طوبولوجية باستخدام جميع LSAs المستقبلة. ثم يتم استخدام طريقة المسار الأقصر لحساب إمكانية الوصول إلى الوجهات المتصلة بالشبكة، ويتم استخدام هذه المعلومات لتحديث جدول التوجيه، ويمكن لهذه العملية اكتشاف التغيرات في طوبولوجيا الشبكة الناجمة عن نمو الشبكة أو فشل عنصر ما فيها.

في الواقع، يتم تشغيل تبادل LSA عن طريق حدث في الشبكة، وذلك بدلاً من تشغيله بشكل دوري، مما يسرع إلى حد كبير من عملية التقارب، لأنه لا داعي لانتظار انتهاء سلسلة من العدادات قبل البدء بتقارب أجهزة التوجيه المتصلة بالشبكة.

5.5.5. فوائد بروتوكولات حالة الوصلة

فيما يلي بعض فوائد بروتوكول توجيه حالة الوصلة:

- عادة ما يكون استكشاف الأخطاء وإصلاحها أسهل في شبكات حالة الوصلة لأن كل جهاز توجيه لديه نسخة كاملة عن طوبولوجيا الشبكة أو على الأقل نسخة عن منطقته الخاصة ضمن الشبكة.
- بروتوكولات حالة الوصلة عادة ما تتوسع إلى شبكات أكبر من بروتوكولات شعاع المسافات، ولا سيما بروتوكولات شعاع المسافة التقليدية مثل RIP. [14]

6.5.5. البروتوكول Open Shortest Path First (OSPF)

هو بروتوكول توجيه بمعايير مفتوحة وقد تم تنفيذه من قبل مجموعة واسعة من مزودي أجهزة الشبكات بما في ذلك سيسكو، ويستخدم هذا البروتوكول خوارزمية ديكسترا، ففي البداية يتم إنشاء بنية شجرية لأقصر المسارات، ثم يتم نشر جدول التوجيه بناءً على أفضل المسارات، حيث يتقارب OSPF بشكل سريع، ربما ليس أسرع من EIGRP، كما أنه يدعم عدة مسارات متساوية باتجاه الهدف.

وقد تم تصميم بروتوكول OSPF ليعمل وفق بنية هرمية، أي العمل على تقسيم الشبكات الكبيرة إلى شبكات أصغر تسمى مناطق (Areas) ، والسبب في إنشاء هذه البنية هو ما يلي :

- التخفيض من الحمل على الشبكات نتيجة عمليات التوجيه.
- تسريع تقارب الشبكات.
- الحد من عدم استقرار الشبكة في مناطق منفردة عوضاً عن تأثيره على كامل الشبكة.

وهذا كله يجعل إعدادات OSPF أكثر تفصيلاً وصعوبةً.

1.6.5.5. مصطلحات البرتوكولOSPF

- Link: عبارة عن شبكة أو بوابة على الموجه تتصل مع الشبكة، وعندما يتم تعريف بوابة في بروتوكول OSPF يتم اعتبارها على أنها وصلة، ويكون لها معلومات عن حالتها إذا كانت تعمل أم لا بالإضافة إلى عنوان الـ IP .

- Router ID: عبارة عن عنوان IP يستخدم لتعريف الموجه، وهو أكبر عنوان IP على بوابات Loopback وفي حال لم يتم إعداد هذا النوع من البوابات فسيختار OSPF أعلى عنوان على البوابات الفعالة.
- Neighbors: الجيران هم موجهَين أو أكثر لديهم بوابة على شبكة مشتركة، كاتصال موجهين مباشرة مع بعضهما عن طريق وصلة تسلسلية.
- Adjacency: هي العلاقة بين موجهين يستخدمان البروتوكول OSPF والتي تسمح بالتبادل المباشر لتحديثات المسار.
- Neighborship database: عبارة عن قائمة من موجهات تستخدم البروتوكول OSPF والتي تمت مشاهدة حزم (Hello) الخاصة بها. وتحتفظ بمجموعة متنوعة من التفاصيل، بما في ذلك معرف جهاز التوجيه ID وحالة الموجه على كل جهاز توجيه في قاعدة البيانات هذه.
- Topology Database: تتضمن معلومات من كل حزم LSA التي تم استقبالها من منطقة ما. ويستخدم الموجه قاعدة البيانات هذه كمدخل لخوارزمية ديكسترا والتي تعمل على حساب أقصر مسار لكل شبكة.
- LSA (Link State Advertisement): عبارة عن حزمة بيانات خاصة بالبروتوكول OSPF والتي تتضمن معلومات عن التوجيه وحالة الوصلة حيث يتم مشاركتها بين مختلف موجهات تستخدم البروتوكول OSPF، ويقوم موجه OSPF بتبادل حزم LSA فقط مع الموجهات التي أنشأ بينها علاقة جوار.
- DR (Designated Router): يتم اختيار جهاز توجيه معين كلما كانت موجهات OSPF متصلة بنفس الشبكة المتعددة النفاذ، وهي الشبكات التي لديها عدة مستقبلين.
- BDR (Backup Designated Router): هو البديل الجاهز لـ DR على شبكة بها عدة وصلات تتلقى BDR، وجميع تحديثات التوجيه من موجهات OSPF المجاورة، ولكن لا تغمر الشبكة بتحديثات LSA.
- OSPF areas: وهي مناطق عمل البروتوكول، حيث أن جميع الموجهات المتواجدة في نفس المنطقة يكون لها نفس رقم المنطقة (Area ID) ، ويمكن للموجه أن يتبع لأكثر من منطقة حيث يمكن أن تكون إحدى البوابات في

المنطقة (1) وباقي البوابات تتبع المنطقة (0) حيث أن كل الموجهات المتواجدة بنفس المنطقة لها نفس الجدول الخاص بالطوبولوجيا.

2.6.5.5. حساب شجرة المسار الأقصر أولاً (SPF tree)

في كل منطقة يحسب كل موجه أفضل/أقصر مسار لكل شبكة في نفس المنطقة، وهذه الحسابات تكون مبنية على المعلومات التي تم جمعها في قاعدة البيانات الخاصة ببنية الشبكة، وتعمل خوارزمية المسار الأقصر أولاً (SPF) على تصور كل موجه للمنطقة ذاتها ضمن بنية شجرية يكون هذا الموجه هو الجذر وتتوضع الشبكات الأخرى وفق الأغصان والأوراق، وبذلك يعمل الموجه على إدخال المسارات في جدول التوجيه.

ويستخدم البروتوكول OSPF مقياساً يشار إليه بالتكلفة، وترتبط التكلفة مع كل بوابة مدرجة في شجرة المسار الأقصر أولاً (SPF)، أما تكلفة المسار بأكمله فهي مجموع تكاليف البوابات ضمن المسار. ويتم حساب التكلفة لكل مسار من أي بوابة فعالة من خلال العلاقة التالية:

10^8 / Bandwidth

على اعتبار أن عرض الحزمة Bandwidth (bps) هي القيمة التي يتم إعطاؤها للبوابة.

وتكون التكلفة الكلية للمسار تراكمية من تكاليف المسارات التي يتكون منها. [14]

7.5.5. خوارزمية شعاع المسافة المحسّن (Enhanced Distance Vector Algorithm)

ويستخدم بروتوكول شعاع المسافات المحسّن، أو بروتوكول التوجيه الهجين، شعاع المسافة بمقاييس أكثر دقة لتحديد أفضل المسارات للشبكات الهدف. ومع ذلك، فإنه يختلف عن معظم بروتوكولات شعاع المسافة وذلك بقيامه بتحديث قاعدة بيانات التوجيه عند تغير الطوبولوجيا بدلاً من التحديثات الدورية. ويتقارب بروتوكول التوجيه هذا بسرعة أكبر، مثل بروتوكولات حالة الوصلة. ومع ذلك، فإنه يختلف عن بروتوكولات حالة الوصلة عن طريق الاقتصاد في استخدام الموارد المطلوبة، مثل عرض الحزمة والذاكرة والمعالج.

1.7.5.5. بروتوكولات شعاع المسافة المحسّن (Enhanced Distance Vector Protocols)

البروتوكول EIGRP (Enhanced Interior Gateway Routing Protocol)

تم تصميمه لمعالجة نقاط الضعف في كل من بروتوكولات شعاع المسافة وبروتوكولات حالة الوصلة، فهو مناسب لكثير من الطوبولوجيا المختلفة، وفي شبكة مصممة تصميماً جيداً، فإن هذا البروتوكول يعد خياراً جيداً ويوفر زمن تقارب سريع للغاية مع الحد الأدنى من الحمل على الشبكة ويعد خياراً شائعاً كبروتوكول توجيه على أجهزة سيسكو. ويجمع هذا البروتوكول بين مزايا بروتوكولات توجيه حالة الوصلة وشعاع المسافة. ويعدّ EIGRP بروتوكول شعاع المسافة المحسّن أو بروتوكول توجيه هجين والذي يتضمن الميزات التالية:

- التقارب السريع: بفضل استخدام خوارزمية (Diffusing Update Algorithm DUAL) لتحقيق التقارب السريع.
- يقوم جهاز التوجيه الذي يستخدم EIGRP بتخزين جميع المسارات الاحتياطية المتاحة للوجهات بحيث يمكنه أن يتكيف بسرعة مع المسارات البديلة، أما في حالة عدم وجود مسار أو مسار احتياطي مناسب في جدول التوجيه المحلي، فإن EIGRP يستفسر من جيرانه لاكتشاف المسار البديل.
- انخفاض استخدام عرض الحزمة: لا يصنع EIGRP تحديثات دورية ولكن بدلاً من ذلك، فإنه يرسل تحديثات جزئية عندما يتغير المسار أو المقياس الخاص بهذا المسار. وعندما تتغير معلومات المسار، تقوم DUAL بإرسال تحديث حول هذه الوصلة فقط بدلاً من الجدول بكامله.
- دعم طبقات الشبكة المتعددة: يدعم EIGRP كلاً من AppleTalk وبروتوكول IP بإصداريه الرابع والسادس وIPX، والتي تستخدم موديولات تعتمد على البروتوكول (PDM).
- التوجيه بدون تصنيف (Classless routing): نظراً لأن EIGRP عبارة عن بروتوكول توجيه بدون تصنيف (Classless)، فإنه يعلن عن قناع التوجيه لكل شبكة وجهة، حيث تمكنه ميزة قناع التوجيه من دعم الشبكات الفرعية المتباعدة عن بعضها وأقنعة الشبكات الفرعية المتغيرة الطول VLSM (Variable Length Subnet Mask).

- حمل أقل على الشبكة: يستخدم EIGRP كلاً من البث المتعدد (multicast) والبث الأحادي (unicast) بدلاً من النشر على الشبكة (broadcast) ونتيجة لذلك، لا تتأثر المحطات النهائية بتحديثات التوجيه والطلبات حول معلومات الطوبولوجيا.
- موازنة الحمل: يدعم البروتوكول EIGRP ميزة موازنة الحمل المقياسي، مما يسمح لمسؤولي الشبكة بتوزيع تدفق حركة البيانات بشكل أفضل في شبكاتهم.
- التلخيص السهل: يمكّن EIGRP مسؤولي الشبكة بإنشاء مسارات موجزة في أي مكان داخل الشبكة بدلاً من الاعتماد على طريقة شعاع المسافة التقليدية لإنجاز تلخيص طريق classful فقط ضمن حدود الشبكة الرئيسية. . [14]

استكشاف الجيران:

قبل بدء أجهزة التوجيه EIGRP بتبادل المسارات مع بعضها البعض فيجب عليها أن تصبح جيران، وهنالك ثلاثة شروط يجب تطبيقها لتأسيس علاقة الجوار:

- تلقي رسالة (Hello) أو تأكيد الاتصال (ACK (Acknowledgement .
- مطابقة أرقام AS.
- مقاييس متطابقة (قيم K).

تميل بروتوكولات حالة الوصلة إلى استخدام رسائل (Hello) لتأسيس علاقة الجوار لأنها عادة لا ترسل تحديثات دورية عن المسار، الأمر الذي يوجب أن تكون هناك آلية للمساعدة على التعرف على الجيران عند اتصال أحد موجهات الجيران الجدد، أو أن أحد الموجهات من الجيران القدامى قد انفصل أو تعطل، و للحفاظ على علاقة الجوار، يجب أن تستمر أجهزة التوجيه EIGRP بتلقي رسائل (Hello) من جيرانها.

إن أجهزة التوجيه التي تستخدم البروتوكول EIGRP التي تنتمي إلى أنظمة مستقلة (AS) مختلفة لا تتشارك تلقائياً بمعلومات التوجيه ولا تصبح جيراناً. ويمكن أن يكون لهذا السلوك فائدة حقيقية عند استخدامها في شبكات أكبر لتقليل كمية معلومات المسار التي يتم نشرها من خلال AS محددة. ولكن المأخذ الوحيد هو أننا قد نضطر

إلى التوزيع بين عدة أنظمة مستقلة (AS) مختلفة بشكل يدوي، والوقت الوحيد الذي يعلن فيه البروتوكول EIGRP عن جدول التوجيه بأكمله هو عندما يكتشف جاراً جديداً حيث يشكل علاقة جوار معه من خلال تبادل حزم (Hello). وعندما يحدث ذلك، يعلن كل الجيران عن جداول التوجيه بأكملها لبعضهم البعض. وبعد أن يتعلم كل منها الطرق إلى جيرانها، يتم نشر التغييرات فقط على جدول التوجيه. وعندما تتلقى أجهزة توجيه EIGRP تحديثات جيرانها، يتم تخزينها في جدول طوبولوجيا محلي. ويحتوي هذا الجدول على جميع المسارات المعروفة من جميع الجيران المعروفين، ويتم اختيار أفضل الطرق منها ووضعها في جدول التوجيه.

يحتفظ كل موجه EIGRP بجدول خاص بالجيران، ويحتوي هذا الجدول على قائمة بموجهات EIGRP المتصلة بشكل مباشر والتي لها علاقة جوار مع هذا الموجه. وأيضاً يحتفظ كل موجه EIGRP بجدول خاص بالطوبولوجيا لإعدادات كل بروتوكول، حيث يحتوي جدول الطوبولوجيا على إدخالات المسارات التي تعلمها الموجه لكل وجهة، فيختار البروتوكول EIGRP أفضل مسار للوجهة من جدول الطوبولوجيا ويضعه في جدول التوجيه.

وتدعى أفضل وجهة في البروتوكول EIGRP بالمسار البديل (Successor Route) بينما يدعى المسار الاحتياطي بالبديل المناسب (Feasible Successor)، ولتحديد أفضل مسار بديل وأفضل بديل مناسب للوجهة، فإن هذا البروتوكول يستخدم المحددات التالية:

المسافة المعلن عنها: وهو مقياس البروتوكول EIGRP الدال عن الموجهات الجيران للوصول إلى شبكة معينة.

المسافة المناسبة: وهي المسافة المعلن عنها لشبكة محددة والتي تعلمها من جار EIGRP بالإضافة إلى المقياس الخاص به للوصول إلى ذلك الجار.

ويقارن جهاز التوجيه جميع المسافات المناسبة للوصول إلى شبكة معينة ثم يختار أقل مسافة مناسبة ويضعها في جدول التوجيه. وتصبح المسافة المناسبة للمسار المختار هي مقياس توجيه البروتوكول EIGRP للوصول إلى تلك الشبكة في جدول التوجيه. وتحتوي قاعدة بيانات طوبولوجيا البروتوكول EIGRP على جميع المسارات المعروفة لكل جار من جيران EIGRP. [14]

8.5.5. الفرق بين بروتوكولات حالة الوصلة وشعاع المسافة

كما ذكرنا سابقاً، إن الفرق الرئيسي بين هذه البروتوكولات هو الخوارزمية التي يقوم عليها كل بروتوكول، وقد أدى هذا الاختلاف إلى مجموعة من الميزات التي تم تطبيقها بطريقة مختلفة لبروتوكولات التوجيه مما يجعل من الصعب تعيين مقارنة شاملة بين بروتوكولات حالة الوصلة وشعاع المسافة. فكما ذكرنا سابقاً، يتعلم بروتوكول شعاع المسافة المسارات ويرسلها إلى الأجهزة الجوار المتصلة مباشرة، وعلى نقيض ذلك، فإن بروتوكولات حالة الوصلة تعلن عن حالة جميع الوصلات (من خلال الحزم المعروفة باسم LSAs) التي تشارك في عملية التوجيه، حيث أن الموجهات الأخرى في المنطقة يمكنها بناء قاعدة بيانات الطوبولوجيا، وذلك لأن كل جهاز لديه المعرفة الكاملة للشبكة، فيتم تجنب وجود حلقات التوجيه تماماً.

وعلاوة على ذلك، في بروتوكولات حالة الوصلة، يحدث فيض LSAs عند بدء التشغيل، وكلما كان هناك تغيير في هيكل الشبكة (كتوقف موجه عن العمل أو انقطاع وصلة)، فإنه في هذه الحالة يتم إرسال التحديث الجزئي مباشرة مع معلومات حول الوصلات المعدلة فقط. وإذا لم يكن هناك أي تغيير، فإن التحديثات تجرى بعد زمن محدد. وعلى النقيض من ذلك، ترسل بروتوكولات شعاع المسافة جدول التوجيه بأكمله كل 30 ثانية بشكل افتراضي. ومع ذلك، فمن المهم الإشارة إلى أن فيض LSAs الأولي في بروتوكولات حالة الوصلة يمكنه الحد بشكل كبير من قدرة الشبكة في إرسال البيانات ويخفض بشكل ملحوظ الأداء عند بدء التشغيل. ومن حيث المقاييس فإن بروتوكولات حالة الوصلة تستخدم المقياس الذي يعتمد التكلفة، والذي يحتوي على معلومات حول قدرة كل وصلة، بينما تستخدم بروتوكولات شعاع المسافة مقياس يستند إلى عدد القفزات. بالإضافة إلى ذلك، في بروتوكولات حالة الوصلة، يمكن تعريف المناطق من أجل تقليل حساب SPF وحجم جداول التوجيه وقواعد البيانات، وهذا يؤدي إلى التقارب بشكل أسرع. بينما هناك قيود هامة في بروتوكولات شعاع المسافة حيث أنها لا يمكن تنفيذها لشبكات فيها أكثر من 15 قفزة. عموماً، يبدو أن بروتوكولات حالة الوصلة هي أفضل من بروتوكولات شعاع المسافة، لكن التنفيذ المستقر لبروتوكول حالة الوصلة هو أمر صعب، أما الذاكرة، واستهلاك وحدة المعالجة المركزية فهي أكبر تماماً، ولذلك، لشبكة فيها عدد قليل من أجهزة التوجيه وطوبولوجيا بسيطة، فإن RIP سوف يلبي الاحتياجات بشكل جيد. [15]

أهم القضايا حول الاختلافات بين بروتوكولات شعاع المسافة وحالة الوصلة يمكن تلخيصها في الجدول التالي:

الجدول (5-1)- مقارنة بين بروتوكولات شعاع المسافة وحالة الوصلة

	حالة الوصلة (Link State)	**شعاع المسافة (Distance Vector)**
الخوارزمية	**ديكسترا (Dijkstra)**	**بيلمان-فورد (Bellman-Ford)**
مشهد الشبكة	معرفة مشتركة وكاملة لطوبولوجيا الشبكة	معرفة الطوبولوجيا من الموجه الجار
حساب أفضل مسار	مبنية على الكلفة (القفزات، عرض الحزمة، التأخير..)	مبنية على أقل عدد من القفزات
التحديثات	تحديثات حالة الوصلة	جدول توجيه كامل
تكرار التحديثات	تحديثات مشغَّلة	تحديثات دورية متكررة
حلقات التوجيه	حسب طريقة بنائها، فلا يمكن أن يحدث حلقات توجيه	بحاجة إلى إجرائيات أخرى لتلافيها
المعالج والذاكرة	استخدام كثيف	استخدام منخفض
البساطة	بحاجة إلى مسؤول شبكة متدرب	بساطة كبيرة

الفصل السادس

دراسة تأثير بروتوكولات التوجيه عند نقل أنواع البيانات المختلفة

Study the impact of routing protocols when transmitting different types of data

1.6. مقدمة لأنواع البيانات:

في العصر الحالي، لم تعد شبكات الاتصالات الحاسوبية مجرد وسيلة لتبادل المعلومات النصية البسيطة، بل أصبحت البنية التحتية الأساسية لعدد هائل من التطبيقات والخدمات التي تمس مختلف جوانب حياتنا اليومية. بدءاً من نقل البيانات النصية التي تشكّل العمود الفقري للتواصل بين الأنظمة والتطبيقات، مروراً بخدمات نقل الصوت التي تمثل القلب النابض لوسائل الاتصال الحديثة عبر تقنيات (VoIP (Voice over Internet Protocol، وصولاً إلى خدمات الفيديو التي أصبحت ركيزة أساسية في العديد من المجالات مثل: التعليم، الترفيه، الأعمال التجارية، والتواصل الاجتماعي .. الخ.

هذا التنوع في طبيعة البيانات المنقولة عبر الشبكات يفرض تحديات متزايدة أمام المهندسين والمتخصصين في مجال تكنولوجيا المعلومات حيث تواجههم مشاكل عديدة يجب عليهم التعامل معها، باعتبار أن كل نوع من البيانات يمتلك خصائص مميزة تتطلب بروتوكولات توجيه ملائمة لضمان وصولها بجودة وكفاءة عالية.

وبناءً على دراسة بروتوكولات التوجيه في الفصل الرابع، سنقوم بدراسة تأثير نوع البيانات (صوت، فيديو، نص) على هذه البروتوكولات مع شرح بسيط لنوع كل منها:

- **البيانات النصية (Text Data):**

  تُعتبر الأكثر شيوعاً وبساطة من حيث متطلبات النقل، إلا أن ضمان موثوقية وصولها بدقة وأمان يظل أمراً أساسياً. وهنا تلعب بروتوكولات التوجيه الداخلية دوراً جوهرياً في تنظيم حركة البيانات وتوزيعها بكفاءة بين الأجهزة والأقسام المختلفة داخل الشبكة، مما يسهم في رفع الأداء العام وتحقيق استقرار الخدمات.

- **بيانات الصوت (Voice Data):**

  مع انتشار خدمات الاتصالات الصوتية عبر الشبكات مثل VoIP، ازدادت الحاجة إلى بروتوكولات قادرة على تقليل التأخير (Latency) وضمان انسيابية تدفق الحزم الصوتية دون انقطاع أو فقدان ملحوظ للجودة. فنجاح تجربة المستخدم في هذه الخدمات يعتمد بشكل مباشر على حسن اختيار البروتوكولات التي تُدار بها حركة المرور داخل الشبكة.

- **بيانات الفيديو (Video Data):**

تُعد الأكثر حساسية من حيث متطلبات النطاق الترددي (Bandwidth) وجودة الخدمة (QoS) فمشاهدة بث فيديو مباشر أو حضور محاضرة عبر الإنترنت يتطلبان أداءً عالياً للشبكة من دون تقطع أو تأخير مزعج. لذا فإن اختيار البروتوكول المناسب لنقل هذه البيانات لا يُعتبر خياراً ثانوياً، بل أساساً لتحقيق جودة مطابقة لتوقعات المستخدمين.

من هنا يتضح أن البروتوكولات ليست مجرد أدوات تقنية، بل هي مكوّن استراتيجي يحدد إلى حد بعيد كفاءة عمل الشبكات في مواكبة متطلبات التطبيقات والخدمات المتنوعة. وعليه، فإن دراسة تأثير بروتوكولات التوجيه وفهم الفروق الجوهرية بينها وعلاقتها بنوع البيانات المنقولة عبر الشبكات يُعد أمراً بالغ الأهمية للمتخصصين الراغبين في تصميم وإدارة شبكات قادرة على تقديم أداء متميز في نقل البيانات النصية والصوتية والمرئية على حد سواء.

لذلك في هذا القسم من البحث قمنا بدراسة وتصميم وتحليل عينة تتألف من 50 شبكة (تم اختيار هذا العدد بشكل تجريبي) بأحجام مختلفة وطوبولوجيا مختلفة وحجم نقل بيانات مختلف، حيث كل شبكة تمت دراستها وفق 3 حالات وذلك لكل نوع من أنواع الخدمات المنقولة عبر الشبكات (صوت، فيديو، ملفات نصية)، وكل حالة من الحالات السابقة تم دراستها 3 مرات وذلك لتطبيق 3 بروتوكولات توجيه مختلفة عند كل شبكة من الشبكات. أي أنه تمت دراسة وتحليل 450 شبكة في هذه القسم من البحث.

2.6. العوامل المؤثرة في أداء بروتوكولات التوجيه في هذه الدراسة:

يمكن أن تؤثر مجموعة من العوامل على أداء بروتوكولات التوجيه ضمن الشبكات المختلفة ومن هذه العوامل:

1.2.6. حجم الشبكة:

يعتبر حجم الشبكة من أبرز العوامل التي تؤثر على اختيار البروتوكول المناسب. ففي الشبكات الصغيرة، قد تكون بروتوكولات التوجيه البسيطة والسهلة التنفيذ مثل RIP خياراً مثالياً في بعض الأحيان، حيث توفر حلاً فعالاً لنقل

البيانات دون تعقيد كبير. بينما في الشبكات الكبيرة والمعقدة، قد يتطلب الأمر استخدام بروتوكولات أخرى مثل OSPF أو EIGRP لإدارة الشبكة بكفاءة وفعالية أكبر.

وأيضاً يختلف حجم جداول التوجيه في الشبكات الكبيرة عن الصغيرة، حيث تكون جداول التوجيه كبيرة ومعقدة في الشبكات الكبيرة نظراً لوجود العديد من الشبكات الفرعية والمسارات المتاحة ضمنها. لذلك يجب على بروتوكولات التوجيه أن تتمكن من التعامل مع هذه الجداول بكفاءة وبالتالي يجب أن تكون عمليات البحث والتحديث سريعة وفعالة والتي تختلف من بروتوكول لآخر.

2.2.6. شكل طوبولوجيا الشبكة (Topology):

يمثل شكل طوبولوجيا الشبكة الطريقة التي يتم فيها توصيل الأجهزة والشبكات الفرعية ببعضها البعض وترتيبها في الشبكة، مثل الشبكات النجمية، والشبكات الحلقية، والشبكات المتشابكة (Mesh) حيث يؤثر شكل طوبولوجيا الشبكة على كفاءة عملية التوجيه ونقل البيانات، فقد تكون بعض البروتوكولات أكثر ملاءمة لبعض الطوبولوجيات من غيرها، حيث أنه في الشبكات المتشابكة فرضاً، قد يكون هناك عدد كبير من الوصلات والعقد (Nodes) في الشبكة الواحدة وهذا يمكن أن يؤدي إلى زيادة في تكرار البيانات الأمر الذي يسبب الحمل على بروتوكولات التوجيه حيث تختلف طريقة التعامل مع هذه الحالة من بروتوكول إلى آخر.

3.2.6. كمية البيانات المرسلة:

تعتبر كمية البيانات المرسلة ضمن الشبكة واحدة من العوامل الرئيسية التي يجب مراعاتها عند اختيار البروتوكول المناسب، فكلما ازدادت كمية البيانات المنقولة ازداد استخدام عرض المجال الترددي واستخدام موارد المعالجة في الأجهزة المختلفة بما فيها استخدام ذاكرة التخزين المؤقتة، بالإضافة إلى أن زيادة كمية البيانات يمكن أن تؤدي إلى زيادة في تأخير الشبكة، خاصة عند تجاوز سعة المجال الترددي المتاح، وأيضاً فإن زيادة كمية البيانات يمكن أن تزيد من معدل الخطأ في الشبكة سواء كانت بسبب الازدحام أو الضجيج أو الأخطاء بعملية النقل. لذلك يجب على بروتوكولات التوجيه أن تكون قادرة على إدارة العوامل السابقة وتحقيق أداء مقبول وذلك يختلف من بروتوكول لآخر حسب الخوارزمية التي يعتمد عليها.

مما سبق نستنتج أن تلك العوامل تتفاعل بشكل معقد مع أداء بروتوكولات التوجيه داخل الشبكات، ولكن عند تقييم تأثير هذه العوامل بشكل صحيح، عندها يمكن تحقيق تحسينات كبيرة في أداء الشبكة مع زيادة فعالية عمليات نقل البيانات. (Dordal. 2017) [16]

3.6. دراسة تأثير بروتوكولات التوجيه عند نقل البيانات النصية

1.3.6. محددات القياس المستخدمة في الدراسة:

في هذا البحث تم دراسة عدة محددات مثل:

- زمن استجابة الاستعلام عن البيانات ضمن قاعدة البيانات
- نسبة ضياع حزم الاستعلام المرسلة
- زمن التأخير الأعظمي لشبكة الإيثرنت
- زمن تقارب الشبكة
- زمن استجابة تحميل الملفات
- نسبة ضياع حزم FTP

وذلك من أجل تحديد أداء بروتوكولات التوجيه عند قياس القيم الناتجة في الشبكات المختلفة. وفيما يلي شرح كل محدد:

1.3.6. 1. زمن تأخير الإيثرنت (Ethernet Delay):

زمن تأخير الإيثرنت يعتمد على عدة عوامل، بما في ذلك نوع الإيثرنت المستخدم والطوبولوجيا الشبكية وحالة الشبكة. ومع ذلك، يمكن تقسيم زمن التأخير في الإيثرنت إلى أربعة أنواع رئيسية:

- زمن تأخير الانتشار (Propagation Delay): هو الزمن اللازم للبيانات لتنتقل عبر وسط النقل، مثل الكابلات النحاسية UTP (Unshielded Twisted Pair) أو كابلات الألياف الضوئية، حيث يتوقف زمن تأخير الانتشار على المسافة بين المرسل والمستقبل.
- زمن تأخير الإرسال (Transmission delay): وهو التأخير بين الأوقات التي يتم فيها إرسال البتات الأولى والأخيرة من الحزمة.

- زمن تأخير الانتظار بالرتل (Queueing Delay): هو الزمن اللازم لانتظار البيانات في طابور الانتظار (Queue) في جهاز التوجيه (Router) أو الجهاز المستلم. قد يحدث هذا التأخير عندما تكون هناك حزم بيانات أخرى في انتظار الإرسال أو الاستقبال.
- زمن تأخير المعالجة (Processing Delay): هو الزمن اللازم لمعالجة البيانات في جهاز التوجيه أو الجهاز المستلم، مثل تحليل العناوين وتحديد طريقة التوجيه. هذا الزمن يمكن أن يختلف اعتماداً على موارد المعالجة المتاحة.

بشكل عام، يكون زمن تأخير الإيثرنت منخفضاً نسبياً، ولكن يمكن أن يزيد بشكل كبير مع زيادة الازدحام في الشبكة أو الحمل على الجهاز الشبكي. (Bertsekas *et al.*, 1992) [17]

1.3.6. 2. ضياع حزم الاستعلام عن البيانات:

يشير ضياع حزم الاستعلام في قواعد البيانات إلى فقدان بعض الحزم أو الطلبات التي تم إرسالها من التطبيق إلى قاعدة البيانات، عبر بروتوكول تدفق البيانات الجدولي (TDS: Tabular Data Stream) العامل في طبقة التطبيقات والمسؤول عن اتصال المستخدمين بمخدم قواعد البيانات، أو الفقدان الجزئي أو الكلي للبيانات المرتجعة منها. ويمكن أن يكون هذا الضياع نتيجة لعدة عوامل وذلك كما يلي:

- ازدحام الشبكة: إذا كان هناك ازدحام في الشبكة بين التطبيق وقاعدة البيانات، فقد تُفقد بعض حزم الاستعلام أو تتأخر في وصولها.
- خلل في الشبكة: قد يتسبب خلل ما في الشبكة بفقدان حزم الاستعلام أثناء النقل بين التطبيق وقاعدة البيانات، مثل فقدان الحزم بسبب تداخل إشارة أو خطأ في عملية التوجيه.
- مشكلات في قاعدة البيانات: قد تحدث مشكلات معينة في قاعدة البيانات نفسها والتي تؤدي لاحقاً إلى فقدان حزم الاستعلام، مثل انهيار قاعدة البيانات أو مشكلات في نظام إدارتها.
- تحميل زائد على قاعدة البيانات: إذا كان هناك تحميل زائد على قاعدة البيانات بسبب العدد الكبير من الاستعلامات أو العمليات المتزامنة، فقد تؤدي هذه الحالة إلى ضياع بعض الحزم أو تأخير في معالجة عمليات الاستعلام.

- مشكلات البرمجة: قد تكون هناك مشكلات في برمجة التطبيق نفسه تؤدي إلى فقدان حزم الاستعلام، مثل خطأ في تنفيذ الطلب أو استخدام بروتوكولات غير موثوقة للتواصل مع قاعدة البيانات.
- مشكلات أخرى: مثل انقطاع الكابلات أو انقطاع التيار الكهربائي.

وتحسب نسبة الضياع بحاصل قسمة عدد الحزم المفقودة على عدد الحزم المرسلة، وكلما كانت هذه النسبة أصغر كان أداء البروتوكول أفضل. (Ramakrishnan *et al.*, 2022) [18]

1.3.6. 3. زمن استجابة الاستعلام في قاعدة البيانات:

يشير إلى الزمن الذي تستغرقه قاعدة البيانات لإرجاع النتائج بعد استلام الاستعلام من التطبيق، ويتأثر زمن الاستجابة بعدة عوامل، منها:

- كمية البيانات: كلما زادت كمية البيانات المطلوب استعلامها، زاد زمن الاستجابة عادةً. وستستغرق عمليات البحث والفرز والتحليل وقتاً أطول.
- الطلب على قاعدة البيانات: إذا كان هناك حمل زائد على قاعدة البيانات، مثل عدد كبير من الاستعلامات المتزامنة أو العمليات المتعددة، فقد يزيد هذا من زمن الاستجابة.
- أداء المخدم (Server): يمكن أن يؤثر أداء المخدم الذي تعمل عليه قاعدة البيانات بشكل كبير على زمن الاستجابة، فالمخدمات ذات الموارد القوية والأداء الجيد يمكن أن تسرع عمليات الاستعلام في قاعدة البيانات.
- شبكة الاتصال: يمكن أن يتأثر زمن الاستجابة أيضاً بسرعة وكفاءة شبكة الاتصال بين التطبيق وقاعدة البيانات، فانخفاض سرعة الشبكة أو ارتفاع الكثافة المرورية للبيانات في الشبكة قد يؤدي إلى تأخير في نقل البيانات. (Ramakrishnan *et al.*, 2022) [17]، (Forouzan. 2012) [19]

1.3.6. 4. زمن التقارب في الشبكة:

وهو الفترة الزمنية التي تحتاجها الشبكة للوصول إلى حالة استقرار بعد بدء تشغيل الأجهزة الشبكية أو عند حدوث تغييرات مثل فشل وصلة أو جهاز توجيه. خلال هذه الفترة، تقوم أجهزة التوجيه بتبادل المعلومات حول الشبكة وتحديث جداول التوجيه لتعكس الطوبولوجيا الجديدة، هذا التحديث يضمن جاهزية جميع الأجهزة لبدء عملية التوجيه بكفاءة

عالية ويمكن أن تكون فترة التقارب قصيرة أو طويلة اعتماداً على البروتوكولات المستخدمة وحجم الشبكة وتعقيدها. بعض البروتوكولات مثلEIGRP مصممة لتحقيق تقارب سريع مما يقلل من تأثير فشل الشبكة ويساعد في الحفاظ على الأداء الأمثل، والوصول إلى حالة استقرار سريع يساهم في تقليل احتمالات فقدان البيانات وضمان استمرارية الاتصال في الشبكة. (Sachinidis. et al., 2023) [20].

1.3.6. 5. زمن استجابة تحميل (File Transfer Protocol) FTP:

ويتم تعريفه على أنه الوقت المنقضي المستغرق لطلب واستقبال الاستجابة عبر بروتوكول نقل الملفات FTP (وهو بروتوكول من بروتوكولات طبقة التطبيقات)، وهذا الزمن يمكن أن يتأثر بعدة عوامل، منها: (Akintola. *et al*., 2024) [21].

- سرعة الاتصال: حيث كلما كانت سرعة الاتصال أعلى، كلما كان زمن الاستجابة أقل.
- حجم الملف: إن الملفات الأكبر حجماً تحتاج إلى زمن أعلى من أجل المعالجة والتحميل، مما يزيد من زمن الاستجابة.
- أداء المخدم: قدرة المخدم على معالجة الطلبات بسرعة تؤثر أيضاً على زمن الاستجابة، فكلما زادت قدرة المخدم على المعالجة كلما نقص زمن الاستجابة.
- الطوبولوجيا والمسافة الجغرافية: طريقة توصيل الشبكة والمسافة بين العميل والمخدم يمكن أن تؤثران على زمن الاستجابة بسبب التأخير الذي قد يحصل في نقل البيانات.

1.3.6. 6. ضياع حزم FTP (بروتوكول نقل الملفات):

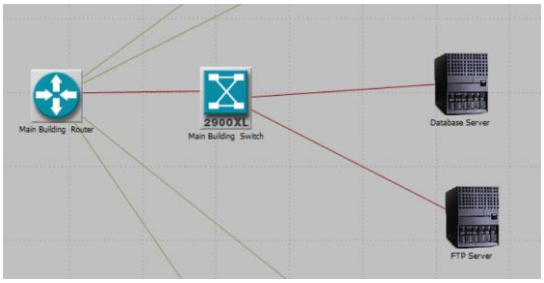
يشير إلى فقدان بعض الحزم أثناء عملية نقل البيانات عبر الشبكة باستخدام بروتوكولFTP ، ويحدث هذا الضياع عندما لا تصل الحزم المرسلة من جهاز إلى آخر بسبب الاصطدامات والازدحام على الشبكة أو بسبب قطع بخطوط الاتصال وغيرها، مما يؤدي إلى عدم اكتمال نقل الملفات أو حدوث أخطاء في البيانات المستلمة. (Kristianto. *et al*., 2021) [22].

2.3.6. تحليل الشبكات التي تم دراستها:

- من أجل الحصول على نتائج أكثر وثوقية، تم القيام بدراسة عدد من الشبكات بأحجام مختلفة وبعدة أشكال من الطوبولوجيا (نجمية، خطية، متشابكة (Mesh)، حلقية، هجينة) بالإضافة لنقل أكثر من حجم بيانات لدراسة أداء بروتوكولات التوجيه بظروف مختلفة، وذلك باستخدام محاكي الشبكات OPNET وبناء خمسين شبكة مختلفة وفق العوامل المذكورة وتطبيق بروتوكولات التوجيه الثلاث (RIP– OSPF – EIGRP) كل بروتوكول بسيناريو مستقل ضمن كل شبكة من هذه الشبكات وتحليل كل واحدة منها بشكل منفرد ومن ثم مقارنة النتائج التي حصلنا عليها لكل بروتوكول توجيه، أي أنه تم في هذه الدراسة إجراء 150 تحليل مختلف.

3.3.6. مواصفات البنية التحتية للشبكات:

- في كل شبكة من الشبكات المدروسة تم توصيل مخدم قواعد البيانات ومخدم الملفات مع المبدل وموجه الشبكة الرئيسيان ضمن الشبكة المركزية، والتي تتصل مع الشبكات الفرعية عن طريق مبدلات وموجهات الشبكات الفرعية وصولاً إلى الحواسب الطالبة لهذه الخدمة.

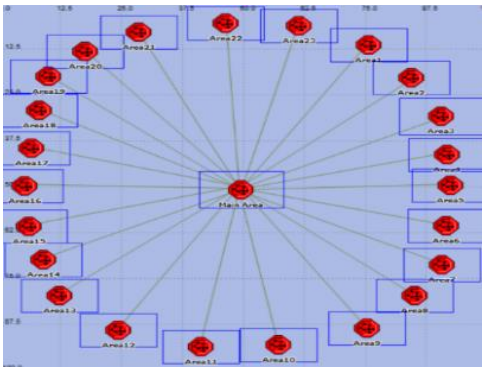


الشكل (6–1)– مكونات الشبكة المركزية

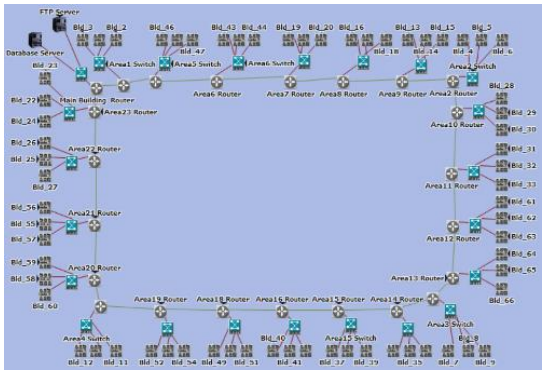
- لقد تم وصل موجهات الشبكة بين بعضها البعض عن طريق كابلات E1، أما وصل حواسب المستخدمين فكان عن طريق كابلات (UTP) بسرعة 100 Mbps.
- في كل شبكة فرعية تم توصيل ثلاثة أبنية وفي كل بناء يوجد (20) حاسب، (10) حواسب منها تعمل كطرفيات لخدمة قواعد البيانات والعشرة الأخرى تعمل كطرفيات لخدمة FTP.

- تم نقل البيانات في الشبكات المدروسة بعدة أحجام مختلفة، الأول كبير نسبياً والثاني متوسط والثالث متدرج بينهما من أجل تشغيل الشبكة بوضعيات مختلفة (أحمال متنوعة) وذلك لدراسة بروتوكولات التوجيه في ظروف نقل بيانات بأحجام مختلفة تطابق الواقع الفعلي في هذه الحالات، وهي:
- الحالة الأولى (حجم بيانات كبير نسبياً) تم تطبيقها على (10) شبكات من الشبكات المدروسة، أما أحجام البيانات فهي كالآتي:
  - حجم كل طلب من طلبات قواعد البيانات 16 KB
  - حجم كل ملف من الملفات المنقولة: 100 KB
- الحالة الثانية (حجم بيانات متوسط نسبياً) تم تطبيقها على (28) شبكة، وهي كما يلي:
  - حجم كل طلب من طلبات قواعد البيانات 7 KB
  - حجم كل ملف من الملفات المنقولة: 10 KB
- الحالة الثالثة (أحجام بيانات متدرجة بين الكبيرة والمتوسطة) تم تطبيقها على (12) شبكة وفق مايلي:
  - الحجم الأول:
    - حجم كل طلب من طلبات قواعد البيانات 13 KB
    - حجم كل ملف من الملفات المنقولة: 80 KB
  - الحجم الثاني:
    - حجم كل طلب من طلبات قواعد البيانات 11 KB
    - حجم كل ملف من الملفات المنقولة: 70 KB
  - الحجم الثالث:
    - حجم كل طلب من طلبات قواعد البيانات 9.6 KB
    - حجم كل ملف من الملفات المنقولة: 60 KB

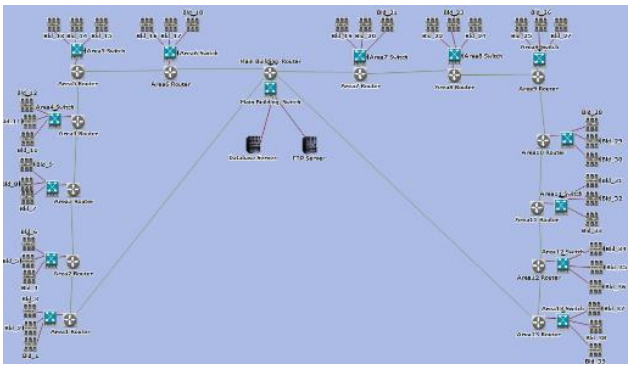
- تمت مقارنة أداء بروتوكولات التوجيه بين بعضها البعض في كل شبكة حيث تم تحليلها بناءً على محددات دراستنا للوصول إلى البروتوكول المناسب لكل شبكة من الشبكات المدروسة.
- المخططات التالية توضح بعض من الشبكات الخمسين المدروسة والتي تختلف بأحجامها وعدد المستخدمين ضمنها بالإضافة لشكل الطوبولوجيا الخاصة بها:



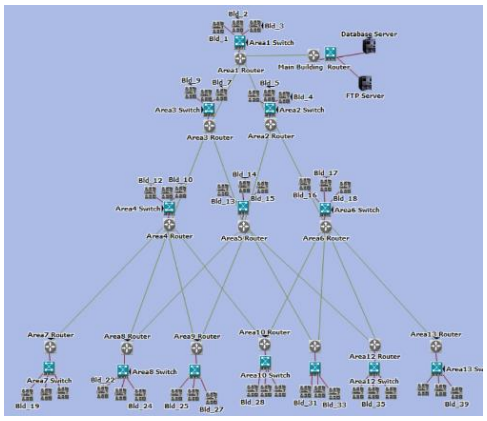
الشكل (6-2)- شبكة نجمية



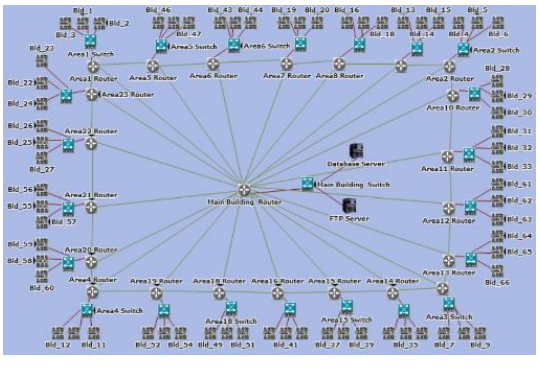
الشكل (6-3)- شبكة حلقية



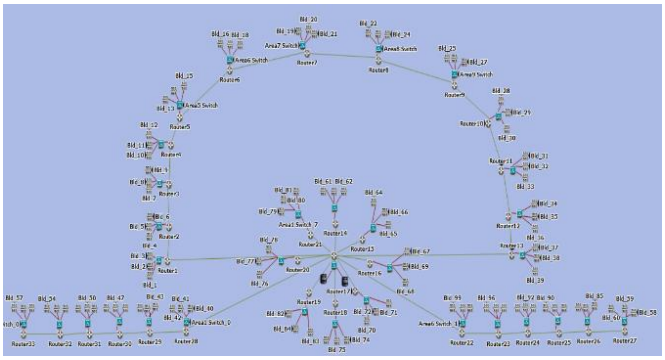
الشكل (6-4)- شبكة هجينة



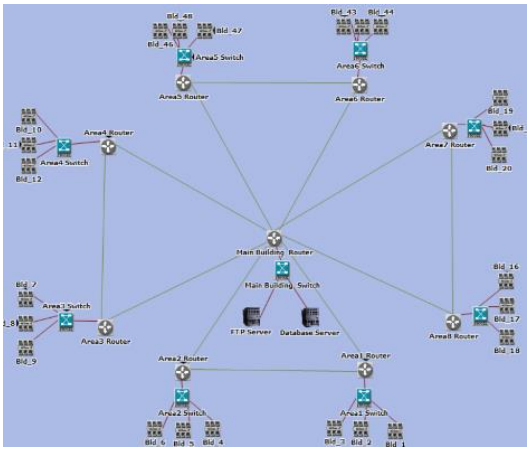
الشكل (6-5)- شبكة هرمية



الشكل (6-6)- شبكة متشابكة (Mesh) كبيرة



الشكل (6-7)- شبكة هجينة كبيرة



الشكل (6-8)- شبكة متشابكة (Mesh) صغيرة



الشكل (6-9)- شبكة خطية

4.3.6. تقييم نتائج التحليل:

تم جمع نتائج تحليل كل شبكة من الشبكات عند تطبيق البروتوكولات الثلاث (كل بروتوكول بسيناريو مستقل) وفق محددات الدراسة في هذا البحث وهي ستة محددات:

- زمن الاستجابة لاستعلام قاعدة البيانات (DB Query Response Time)
- نسبة الضياع لحزم استعلام قاعدة البيانات المرسلة (DB Query packet's loss rate)
- زمن التأخير الأعظمي لشبكة الإيثرنت (Ethernet Delay)
- زمن تقارب الشبكة (Convergence Duration)
- زمن استجابة تحميل الملفات (FTP Download Response Time)
- نسبة ضياع حزم FTP (FTP packet's loss rate)

فوجدنا أن الفارق في قيم المحدد (تقارب الشبكة) كان كبيراً في جميع الشبكات لصالح بروتوكول التوجيه EIGRP، أما فيما يخص باقي المحددات المدروسة (زمن التأخير، زمن الاستجابة، معدل ضياع الحزم) للخدمتين (قواعد البيانات و FTP)، فنجد تقارب في القيم للبروتوكولات الثلاث في بعض الشبكات واختلافها بالبعض الآخر، حيث كانت هذه القيم كبيرة في عدة شبكات تبعاً لحركة مرور الحزم والازدحام الحاصل ضمنها وحسب طوبولوجيا الشبكة المدروسة ونتيجةً أيضاً لحجم البيانات الكبير المنقول فيها، ونلاحظ انخفاضها في البعض الآخر وخاصة عند انخفاض حجم هذه البيانات، فقد تم تطبيق أحجام ذات قيم مختلفة ومتدرجة بين المتوسطة والكبيرة وإعطاء قيمة من هذه القيم لكل شبكة

من الشبكات المدروسة، فبالنسبة لقواعد البيانات فإنه تم تطبيق أحجام البيانات التالية للطلبات وذلك كما يلي:
16 KB, 13 KB, 11 KB, 9.6 KB, 7 KB، أما بالنسبة لخدمة FTP فتم تطبيق الأحجام التالية للملفات المنقولة:
100 KB, 80 KB, 70 KB, 10 KB.

لقد وجدنا ازدياد قيم المحددات السابقة الذكر بشكل طردي مع زيادة حجم البيانات بالإضافة إلى تغير هذه الزيادة بحسب طوبولوجيا الشبكة المستخدمة.

ففي شبكات الطوبولوجيا الحلقية: نجد أن هذه القيم ازدادت بشكل ملفت عند زيادة متوسطة بحجم البيانات المنقولة ضمنها، ويعود ذلك بسبب انتقال الحزم من المخدم إلى الطرفية عبر موجهات الشبكة بشكل متتالي وعن طريق كابلات E1 ذات سرعة نقل البيانات المحدودة، أضف إلى ذلك تكوّن أحجام أرتال انتظار كبيرة في التجهيزات الموجودة داخل الشبكة وخاصة عندما تكون كتل البيانات كبيرة، مما يتطلب وقتاً أطول للمعالجة، وزيادة زمن الانتظار في الأرتال والذي يسبب لاحقاً حدوث اختناقات في مرور حزم البيانات وبطء في الأداء بشكل عام.

بينما في الشبكات ذات الطوبولوجيا النجمية: فنلاحظ كفاءة هذه الشبكات مقارنة بالشبكات ذات أنواع الطوبولوجيا الأخرى عند نفس حجم البيانات المنقولة، حيث يتم فيها نقل البيانات من المخدم إلى الطرفيات بشكل مركزي وبشكل أسرع من باقي الأنواع المدروسة مما يمكّنها من نقل أحجام بيانات أكبر وبسلاسة أكثر دون حدوث ضياعات مماثلة لتلك التي تحدث في أنواع الشبكات الأخرى.

أما الشبكات المتبقية ذات الطوبولوجيا (الهرمية والخطية والهجينة والمتشابكة): فأتت بأداء يتراوح بين الحلقية والنجمية من حيث حجم البيانات التي يمكنها نقله بأقل نسبة ضياع ممكنة، ولكن الشبكات ذات الطوبولوجيا المتشابكة كانت الأفضل في هذه الناحية.

لذلك نجد أن حجم وكمية البيانات المرسلة ضمن الشبكة بالإضافة لشكل بنية هذه الشبكة تؤثر بشكل فعّال على أدائها، مما يتطلب آليات للتعامل مع الشبكات بمختلف الظروف، منها استخدام تقنيات مثل ضغط البيانات أو تقسيمها إلى كتل أصغر، وتحسين خوارزميات إدارة الأرتال مما يساعد في تقليل التأخيرات الناتجة عن أحجام كتل البيانات الكبيرة وتقليل الاختناقات، كما أن اختيار بروتوكول التوجيه المناسب يمكن أن يؤدي إلى نقل هذه البيانات بكفاءة أعلى

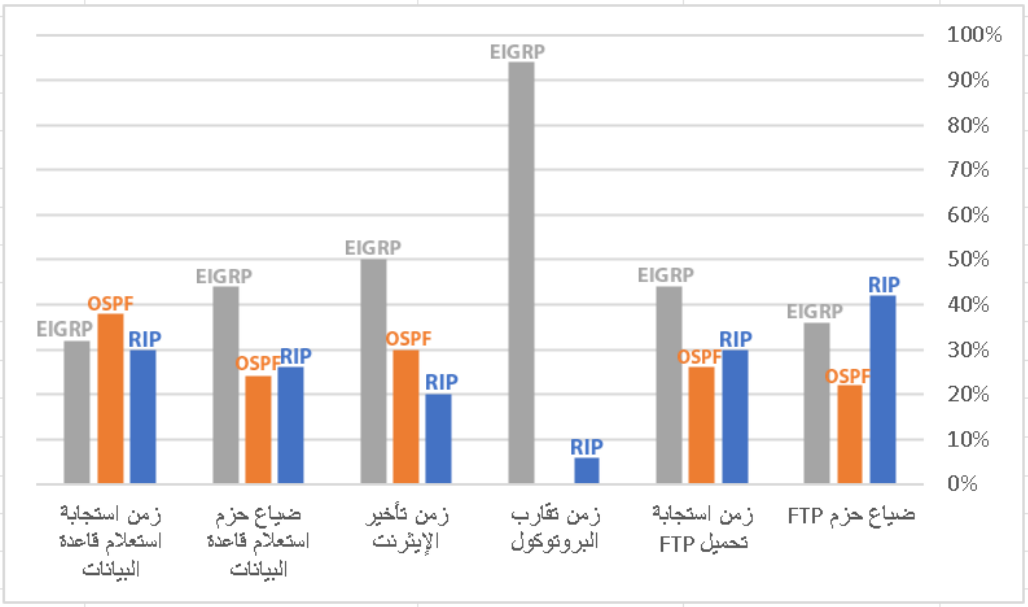
وتحسين الأداء.

تم إيجاد نسبة عدد المرات التي كان فيها أداء كل بروتوكول من بروتوكولات التوجيه أفضل من البروتوكولين الآخرين وفقاً لمحددات الدراسة المعتمدة وذلك بحاصل قسمة عدد المرات التي كان فيها البروتوكول أفضل بالأداء من البروتوكولات الأخرى على عدد الشبكات الكلي (50 شبكة)، كما بالجدول (6-1):

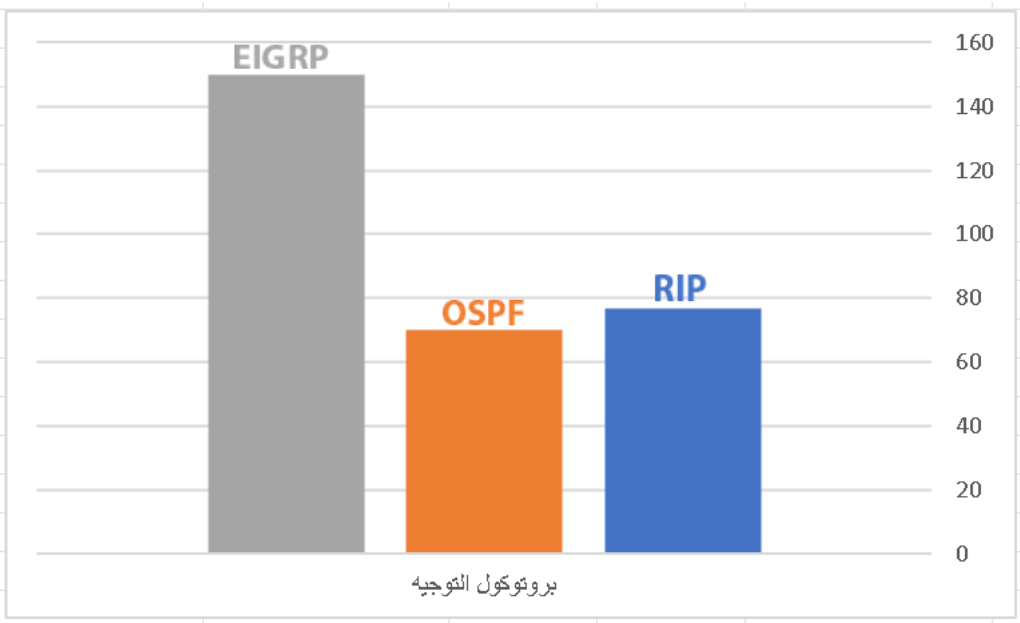
الجدول (6-1)- نسبة عدد المرات التي كان فيها كل بروتوكول أفضل أداءً عند نقل البيانات النصية

	نسبة عدد المرات التي كان فيها البروتوكول أفضل بزمن استجابة استعلام قاعدة البيانات	نسبة عدد المرات التي كان فيها البروتوكول أفضل بضياع حزم استعلام قاعدة البيانات	نسبة عدد المرات التي كان فيها البروتوكول أفضل بزمن تأخير الإيثرنت	نسبة عدد المرات التي كان فيها البروتوكول أفضل بزمن تقارب الشبكة	نسبة عدد المرات التي كان فيها البروتوكول أفضل بزمن استجابة تحميل FTP	نسبة عدد المرات التي كان فيها البروتوكول أفضل بضياع حزم FTP
RIP	30%	26%	20%	6%	30%	42%
OSPF	38%	24%	30%	0%	26%	22%
EIGRP	32%	44%	50%	94%	44%	36%

ويمكن تمثيل نتائج الجدول (6-1) بيانياً كما بالشكلين (6-10) و (6-11):



الشكل (6-10)- نسبة عدد المرات التي حققت فيها بروتوكولات التوجيه أفضل أداء لكل محدد من محددات الدراسة عند نقل البيانات النصية



الشكل (6-11)- عدد المرات الكلي التي حقق فيها بروتوكول التوجيه أفضل أداء بالنسبة لمحددات الدراسة عند نقل البيانات النصية

من الجدول (6-1) والشكلين (6-10) و (6-11) نستنتج تفوق البروتوكول EIGRP بعدد المرات التي كان فيها أفضل من البروتوكولين OSPF وRIP في أربعة محددات ضمن الدراسة، بينما تفوق البروتوكولين الآخرين وبنسبة مقاربة للبروتوكول EIGRP في المحددين المتبقيين، حيث حقق البروتوكول EIGRP أفضل أداء في نسبة ضياع حزم استعلام قاعدة البيانات وبنسبة 44% من الشبكات المدروسة بينما جاء تالياً البروتوكولين RIP وOSPF وبنسبتين متقاربتين حيث حققا بالترتيب النسب 26% و24% من الشبكات المدروسة.

أما فيما يخص المحدد الثاني (زمن تأخير الإيثرنت) فقد حصل البروتوكول EIGRP على أعلى نسبة وهي 50% من الشبكات المدروسة فيما حصل تالياً البروتوكول OSPF على نسبة 30% بينما حصل البروتوكول RIP على نسبة 20% من الشبكات المدروسة.

أما بالمحدد الثالث وهو (زمن تقارب الشبكة)، فقد حقق البروتوكول EIGRP الأداء الأفضل بلا منازع (حيث من المعروف سرعة تقارب شبكة البروتوكول EIGRP) فقد كانت نسبة عدد المرات التي حصل فيها على أفضل أداء من البروتوكولين الآخرين هي 94% من الشبكات المدروسة، فيما حصل البروتوكول RIP على النسبة 6%، بينما لم يتفوق البروتوكول OSPF بأي شبكة على البروتوكولين الآخرين في هذا المحدد وهذا يعود إلى أن البروتوكول OSPF

يغمر الشبكة بمعلومات عن طوبولوجيا الشبكة عند بدء التشغيل من أجل تكوين صورة كاملة للشبكة في كل جهاز توجيه مما يسبب زيادة زمن التقارب الخاص بالشبكة.

وفيما يتعلق بالمحدد الرابع الذي تفوق فيه البروتوكول EIGRP وهو (زمن استجابة تحميل FTP)، فقد حصل على نسبة 44%، بينما حصل البرتوكولين RIP وOSPF على النسبتين 30% و26% على التوالي.

أما المحدد الخامس (زمن استجابة استعلام قاعدة البيانات) فقد حصلت البروتوكولات الثلاث على نسب متقاربة نوعاً ما، حيث نال البروتوكول OSPF على أعلى نسبة وهي 38% بينما جاء تالياً البروتوكول EIGRP وبنسبة 32% وكانت نسبة البروتوكول RIP تساوي 30%.

وأما المحدد السادس والأخير (نسبة ضياع حزم FTP)، فقد كان المركز الأول من نصيب البروتوكول RIP حيث حقق نسبة 42% من الشبكات المدروسة وبنسبة قريبة من EIGRP الذي حقق 36% بينما حصل البروتوكول OSPF النسبة الأقل وهي 22% في هذا المحدد.

لذلك، نستنتج مما سبق أن البروتوكول EIGRP قد تفوق على البروتوكولين OSPF وRIP في أربعة محددات من أصل ستة من محددات الدراسة، وكان مقارباً في الأداء من البروتوكول OSPF الذي تفوق في المحدد الخامس، ومقارباً في الأداء أيضاً من البروتوكول RIP الذي تفوق في المحدد السادس، فكان البروتوكول EIGRP بشكل عام هو البروتوكول الأفضل أداءً في شبكات العينة المدروسة في هذا البحث عند نقل البيانات النصية ضمنها.

هذه الدراسة قامت بتوضيح أثر تغير بروتوكول التوجيه على أداء الشبكة في حالات مختلفة، حيث وجدنا تغير قيم محددات هذه الدراسة عند تشغيل بروتوكولات توجيه مختلفة ضمن الشبكات المدروسة، واتضح لنا بعد ذلك أن أفضل أداء لهذه البروتوكولات كان أداء البروتوكول EIGRP عند نقل البيانات النصية.

4.6. دراسة تأثير بروتوكول التوجيه عند نقل الفيديو:

1.4.6. محددات القياس المستخدمة في الدراسة:

لقياس أداء البروتوكولات، تمت دراسة عدة محددات وهي:

- زمن تقارب البروتوكول
- نسبة ضياع حزم الفيديو
- زمن تأخير حزم الفيديو من طرف إلى طرف
- تباين تأخير الحزم

إن تحليل هذه المحددات يساعد على فهم كيفية أداء البروتوكولات في سيناريوهات مختلفة، وذلك كما يلي:

1.1.4.6. زمن تقارب البروتوكول في الشبكة: قد تم شرحه سابقاً في الفقرة (4 .1.3.5).

2.1.4.6. ضياع حزم الفيديو: يحدث الضياع في حزم الفيديو المستقبلة عند فشل بعض الحزم المرسلة عبر الشبكة بالوصول إلى الوجهة، يمكن أن يكون فقدان الحزم عند جهاز التوجيه وذلك نتيجة تجاهل بعض الحزم على وصلة خرج الجهاز خلال فترة زمنية معينة، مما ينتج عنه فقدان للحزم من نهاية إلى نهاية. حيث تؤدي خسارة الحزم إلى خفض الإنتاجية لمرسل معين، سواء كان ذلك بسبب عطل في الشبكة، أو كوسيلة لتحقيق التوازن بين عرض النطاق الترددي المتاح بين عدة مرسلين عند وصول وصلة جهاز توجيه أو شبكة معينة إلى أقصى قدرة لها. [23] (Sommers J *et al.*2005).

وتحسب نسبة ضياع الحزم بحاصل قسمة عدد الحزم المفقودة على عدد الحزم المرسلة، حيث يحسب عدد الحزم المفقودة بطرح عدد الحزم المستقبَلة من عدد الحزم المرسلَة، وكلما كان ناتج هذه النسبة أصغر كان أداء البروتوكول أفضل.

3.1.4.6. زمن تأخير حزم الفيديو من طرف إلى طرف: إن التأخير من البداية إلى النهاية لكل حزمة هو مجموع التأخيرات التي تواجهها في سلسلة من العقد الوسيطة بطريقها إلى الوجهة. كل تأخير هو مجموع جزأين، جزء ثابت مثل تأخير الإرسال وتأخير الانتشار، وجزء متغير مثل تأخير المعالجة وتأخير الانتظار في العقد. ويؤثر ضياع الحزم

الناجم عن التأخير الكبير من طرف إلى طرف على نوعية الفيديو من جهة المستقبل. (Zhang T *et al*.2019) [24]

4.1.4.6. تباين تأخير حزم الفيديو: وهو الفروق في التأخيرات من طرف إلى طرف لحزم الفيديو، حيث يقاس التأخير من طرف إلى طرف للحزمة من وقت إنشائها إلى وقت استقبالها، وكما أنه يجب استلام بعض الأطر المشفرة في الوقت المناسب لعرضها، في حين يجب استلام البعض الآخر في وقت مبكر لفك تشفير الأطر الأخرى، ولكن في الشبكات التي بها اختلافات كبيرة في التأخير، يمكن أن تصل أطر الفيديو المشفرة متأخرة إلى جهاز فك التشفير وتفوّت مواعيد عرضها الخاصة. ومع ذلك، حتى إذا وصل إطار مشفر بعد وقت التشغيل الخاص به، فإنه لا يزال مفيداً لفك تشفير الأطر اللاحقة.

وفي خدمة المؤتمرات المرئية لا يمكن تجاهل تأخر الحزم بسبب تباين التأخير في قيمها والذي يمكن أن يؤدي إلى فقدانها بالإضافة إلى عدم وجود عملية استرجاع للفيديو. (Wee S. *et al*.2002) [25]

2.4.6. الشبكات المدروسة في البحث:

- تناولت هذه الدراسة اختلاف أداء الشبكة باختلاف بروتوكول التوجيه المطبق ضمنها وذلك عند تغير عدد من العوامل المؤثرة وهي:
  - حجم الشبكة
  - شكل الطوبولوجيا الخاصة بها
  - حجم البيانات المرسلة ضمنها
- في هذه الدراسة تم تحليل عمل 50 شبكة مختلفة بإجراء محاكاة لها باستخدام المحاكي OPNET والذي تتم بواسطته محاكاة بيئات الشبكات الكبيرة مع أجهزة الشبكة وبروتوكولات التوجيه دون الحاجة إلى العمل على الأجهزة الحقيقية، حيث يعتبر هذا المحاكي برنامجاً موثوقاً عندما يتعلق الأمر بتقييم الشبكة.

- لقد تم وصل الشبكة الرئيسية التي تتألف من مخدم خاص بخدمة الفيديو ومبدل رئيسي بالإضافة لموجه شبكة مركزي، ومن ثم وصل هذه الشبكة مع الشبكات الفرعية عن طريق الموجهات الفرعية وصولاً إلى الطرفيات التي تستفيد من هذه الخدمة عن طريق المبدلات الفرعية، أما الخدمات المدروسة في هذه الشبكة فهي خدمة بث الفيديو.
- تم وصل الموجهات في الفروع مع الموجه الرئيسي عن طريق دارات E1.
- تم وصل الطرفيات مع المبدلات الشبكية ومن ثم مع الموجهات لكل فرع عن طريق كابلات إيثرنت بسرعة 100 Mbps.
- تمت مقارنة أداء كل بروتوكول توجيه بشكل منفرد في كل محاكاة على نفس الشبكة المدروسة (أي تمت دراسة كل شبكة ثلاث مرات، مرة لكل بروتوكول توجيه، فتم إجراء (150) محاكاة للوصول للنتائج في هذه الدراسة).
- تم تغيير حجم أطر الفيديو التي يتم نقلها عبر الشبكات المدروسة وبحجم يتراوح بين (2KB-22 KB) لكل إطار وبمعدل 10 frame/sec.
- تم إجراء محاكاة لمدة (20) دقيقة، وهو زمن كافٍ للحصول على نتائج جيدة في معظم الأبحاث في هذا المجال.

تم إنشاء شبكات مختلفة بطوبولوجيا وأحجام مختلفة وتم إنشاء كل نوع من هذه الشبكات بأحجام مختلفة وبعدد مستخدمين مختلف كما بالأشكال (6-2) حتى (6-9) ولكن بمخدم فيديو وطرفيات خاصة به ضمن الشبكة.

3.4.6. جمع النتائج:

تم جمع نتائج تحليل جميع الشبكات عند تشغيل بروتوكولات التوجيه الثلاث ضمن هذه الشبكات فحصلنا على القيم التي حصل عليها كل بروتوكول من بروتوكولات التوجيه المدروسة في كل شبكة عند دراسة المحددات التالية:

- تباين تأخير حزم الفيديو (Video conference packet delay variation)
- زمن تأخير حزم الفيديو من طرف إلى طرف (video conference Packet end-to-end delay)
- نسبة ضياع حزم الفيديو (Video conference packets loss rate)
- وزمن تقارب البروتوكول (Protocol Convergence Duration).

4.4.6. تقييم وتحليل نتائج الدراسة:

في هذه الدراسة لابد لنا من ذكر ملاحظتين لتوضيح عملية تحليل البيانات ضمن الشبكات المدروسة:

أولاً– كان لابد في هذه الدراسة من تغيير حجم الأطر المرسلة ضمن الشبكات المختلفة المدروسة، حيث لبنية الشبكة تأثير كبير على أدائها وعلى عملية نقل البيانات فيها، فكانت الشبكات ذات الطوبولوجيا النجمية والمتشابكة لها الأداء الأفضل من حيث القدرة على نقل بيانات بأحجام كبيرة وبأداء عالي، فيما كانت الشبكات الأخرى مثل الخطية والحلقية ذات قدرة أقل على نقل مثل هذا الحجم من البيانات وذلك بسبب التوصيل المتتالي للأجهزة الشبكية والسرعة المحدودة بنقل البيانات لدارات E1 الواصلة بين الموجهات في شبكات WAN بجميع الشبكات التي تمت دراستها، فينشأ عن ذلك تكوّن أرتال انتظار ضخمة من البيانات عند الأجهزة الشبكية بانتظار المعالجة والنقل وخاصة مع أطر الفيديو الكبيرة، مما يسبب حصول اختناقات ومن ثم زيادة الضياع والذي يمكن أن يؤدي لاحقاً إلى فشل الشبكة.

لذلك تم تدارك هذا الأمر في هذه الدراسة، فتم خفض حجم البيانات المرسلة إلى حد يمكن لهذه الشبكات من نقل البيانات دون الوصول إلى حالة الفشل، حيث كان حجم الأطر في الشبكات ذات الطوبولوجيا النجمية والمتشابكة يصل إلى (22 KB)، فيما وصل حجم الأطر الأقصى في الشبكات ذات الطوبولوجيا الحلقية إلى (7.5 KB) فقط، ومع ذلك وصلت قيمة (تباين تأخير حزم الفيديو) في هذه الشبكات إلى أرقام كبيرة مقارنة مع الشبكات الأخرى، حيث وصلت في بعض الشبكات الحلقية إلى (1083.5 ms) فيما كان وسطي هذا المعامل (136 ms) في الشبكات ذات الطوبولوجيا النجمية والمتشابكة رغم كبر حجم الأطر بداخلها.

ثانياً– يجب الأخذ بعين الاعتبار فروق التأخيرات الطفيفة بين نتائج البروتوكولات عند إجراء محاكاة لشبكة لمدة (20) دقيقة، حيث يمكن لهذه التأخيرات أن تتراكم وتؤثر بشكل كبير على أداء الشبكة مع مرور الزمن وتحديداً في تطبيقات الزمن الحقيقي كما في دراستنا هذه حيث يتم استخدام اتصالات المؤتمرات المرئية، وخاصة إذا كانت هذه الفروق متكررة وعلى مدى فترات طويلة، حيث أن هذا التباين يصبح ملحوظاً في التطبيقات الحساسة للزمن، مما يؤدي إلى تشويش وفقدان جودة الفيديو بالشبكات التي يزداد فيها التأخير وخصوصاً ضمن البيئات التي تعاني من ظروف شبكة غير مستقرة.

1.4.4.6. تحليل الشبكات بدلالة حجم الأطر:

1.1.4.4.6. عند إرسال أحجام الأطر الكبيرة ولطوبولوجيا شبكات مختلفة:

نلاحظ في جميع الشبكات المدروسة بجميع أشكالها تفوق البروتوكول EIGRP على البروتوكولين الآخرين بلا منازع عند قياس تقارب الشبكة، حيث أن هذا البروتوكول معروف بتقاربه السريع ضمن الشبكات، أما في باقي المحددات فنجد أنه في الشبكات النجمية وعند أحجام أطر كبيرة (22 KB) كان أداء بروتوكول التوجيه OSPF أفضل من البروتوكولين الآخرين في المحددات الثلاث (تباين تأخير حزم الفيديو، زمن تأخير حزم الفيديو من طرف إلى طرف، نسبة ضياع حزم الفيديو)، وكذلك الأمر نجد في الشبكات ذات الطوبولوجيا المتشابكة وعند نفس حجم الأطر السابق أن أداء بروتوكول التوجيه OSPF أيضاً كان أفضل من البروتوكولين الآخرين في المحددات الثلاث المذكورة سابقاً.

2.1.4.4.6. عند حجم أطر متغير لنفس الطوبولوجيا الشبكية:

في الطوبولوجيا الحلقية: تم زيادة حجم الأطر من (2 KB) حتى (7.5 KB) ورأينا أن أداء البروتوكول RIP هو الأفضل ضمن شبكة حلقية ذات أحجام أطر صغيرة، ومن ثم تنافس البروتوكولين OSPF وEIGRP بالأداء وبشكل متساوي عند زيادة حجم الأطر المرسلة بداخلها وصولاً لحجم أطر يساوي (7.5 KB).

أما في الطوبولوجيا الهجينة: ومع تغير حجم الأطر، وجدنا أن أداء بروتوكول التوجيه OSPF كان أفضل من البروتوكولين الآخرين وبفارق بسيط عن البروتوكول EIGRP عند دراسة ثلاثة محددات (تباين تأخير حزم الفيديو، زمن تأخير حزم الفيديو من طرف إلى طرف، نسبة ضياع حزم الفيديو).

وفي الطوبولوجيا الخطية: وجدنا أن البروتوكول EIGRP أفضل أداءً في الشبكة الخطية ذات حجم الأطر الأعلى (7.5 KB)، ولكن كان أداء OSPF الأفضل في باقي هذه الشبكات عند أغلب محددات الدراسة.

عند الطوبولوجيا الهرمية: تنوعت النتائج بين البروتوكولات المدروسة، فشبكات كان بها البروتوكول OSPF الأفضل، وشبكات أخرى كان EIGRP هو الأفضل وشبكات تنوعت بها نتائج قيم المحددات بين البروتوكولات الثلاث.

ويمكن تلخيص النتائج عند تغير حجم الأطر كما بالجدول (6-2) التالي:

الجدول (6-2)- علاقة حجم الأطر المرسلة مع أداء بروتوكولات التوجيه حسب الطوبولوجيا عند نقل الفيديو

شكل الطوبولوجيا	الحلقية		الهجينة		الخطية		الهرمية	
حجم الأطر / البروتوكول	صغيرة	متوسطة	صغيرة	متوسطة	صغيرة	متوسطة	صغيرة	متوسطة
RIP	X							
OSPF		X	X	X	X		X	X
EIGRP		X				X	X	X

حيث يوضح الجدول السابق البروتوكول ذو الأداء الأفضل عند كل نوع طوبولوجيا للشبكة وحسب حجم الأطر المرسلة، حيث تمت الإشارة بـ (x) على البروتوكول ذو الأداء الأفضل داخل الأنواع المختلفة لطوبولوجيا الشبكة.

2.4.4.6. تحليل تباين تأخير حزم الفيديو في الشبكات:

1.2.4.4.6. تباين التأخير في الشبكات ذات الطوبولوجيا النجمية:

كانت قيمة هذا المحدد في الشبكات النجمية الصغيرة تقارب (133 ms) للبروتوكول RIP و (132 ms) للبروتوكول OSPF و (136 ms) للبروتوكول EIGRP وعند زيادة حجم الشبكة وتعقيدها وصلت قيمة هذا المحدد إلى (141 ms) للبروتوكول RIP و (136 ms) للبروتوكول EIGRP بينما حافظ البروتوكول OSPF على القيمة ذاتها (132 ms) وهي أفضل من قيم البروتوكولين الآخرين.

2.2.4.4.6. تباين التأخير في الشبكات ذات الطوبولوجيا المتشابكة:

حافظ البروتوكول OSPF في الشبكات ذات الطوبولوجيا المتشابكة على قيمة تقارب (133 ms) وأيضاً حافظ البروتوكول EIGRP على قيم قريبة من (136 ms) فيما ارتفعت قيم البروتوكول RIP من (133 ms) إلى (161 ms) في بعض الشبكات فكان البروتوكول OSPF أفضل من البروتوكولين الآخرين أيضاً في هذا المحدد.

3.2.4.4.6. تباين التأخير في باقي الشبكات:

وفي الشبكات الهجينة والهرمية وجدنا تفوق البروتوكول RIP في الشبكات الصغيرة وذات حجم الأطر الصغير عن البروتوكولين الآخرين، أما عند زيادة حجم الشبكة وتعقيدها فوجدنا تدهور أداء هذا البروتوكول بشكل ملحوظ لتصل قيمة هذا المحدد في بعض الشبكات إلى (1469 ms) أي أكبر بخمسة أضعاف تقريباً من قيمتَي البروتوكولين

الآخرين، وبتحليل بقية الشبكات نجد تفوق البروتوكول OSPF في أغلب شبكات هذا النوع حسب هذا المحدد أيضاً وخصوصاً كلما زاد حجم الشبكة وتعقيدها.

3.4.4.6. تحليل نسبة الضياع في الشبكات:

1.3.4.4.6. نسبة الضياع في الشبكات ذات الطوبولوجيا النجمية:

في الشبكات النجمية الصغيرة كانت نسبة الضياع تصل إلى (15%) للبروتوكولين RIP وOSPF فيما كانت تقارب (17%) للبروتوكول EIGRP، ومع زيادة حجم الشبكة ازدادت نسبة الضياع للبروتوكول RIP لتصل إلى نسبة تقارب (20%) فيما حافظ البروتوكول EIGRP على نفس النسبة (17%) كما حافظ البروتوكول OSPF أيضاً على نسبته السابقة (15%)، فكان أداء البروتوكول OSPF أفضل في هذا النوع من الشبكات حسب هذا المحدد.

2.3.4.4.6. نسبة الضياع في الشبكات ذات الطوبولوجيا المتشابكة:

وكذلك الأمر بالنسبة للشبكات ذات الطوبولوجيا المتشابكة حيث حافظ البروتوكولين OSPF وEIGRP على نسبة ضياع ثابتة وتساوي تقريباً وعلى التتالي (15%) و (17%)، أما البروتوكول RIP فكانت نسبة الضياع في الشبكات الصغيرة تساوي تقريباً (15%)، ولكن عند زيادة حجم الشبكة ازدادت هذه النسبة لتصل في بعض الحالات إلى (87%)، فكان أداء البروتوكول OSPF أفضل أيضاً بهذه النوع من الشبكات.

3.3.4.4.6. نسبة الضياع في الشبكات ذات الطوبولوجيا الهجينة والهرمية:

أما في باقي الشبكات (الهجينة والهرمية) فقد كانت قيم الضياع للبروتوكولين OSPF وEIGRP متقاربة نوعاً ما في بعض الشبكات ولكن كان البروتوكول OSPF أقل بهذا المحدد في شبكات أخرى، بينما كانت قيم البروتوكول RIP مرتفعة نسبياً عنهما لتصل في أغلب الحالات إلى قيم تقارب ضعفي قيم البروتوكول OSPF.

4.3.4.4.6. نسبة الضياع في الشبكات ذات الطوبولوجيا الحلقية:

أما في الشبكات ذات الطوبولوجيا الحلقية فكانت النسب متفاوتة بين انخفاض وارتفاع للبروتوكولات الثلاث.

4.4.4.6. تحليل زمن التأخير من طرف إلى طرف في الشبكات:

في حين وصل الفرق في بعض الشبكات إلى (1307 ms) حيث يعتبر فرق تأخير كبير وخاصة في الشبكات الضخمة، نجد أنه في هذا المحدد كانت القيم كبيرة بشكل عام، وتقاربت قيم البروتوكولات الثلاث في أغلب الشبكات بفروق قليلة، ولكن وكما ذكرنا سابقاً، يجب أخذ فرق التأخيرات الطفيفة بين البروتوكولات المدروسة أيضاً بعين الاعتبار عند إجراء محاكاة لشبكة لمدة (20) دقيقة فقط، حيث يمكن لهذه التأخيرات أن تتراكم وتؤثر بشكل كبير على أداء الشبكة مع مرور الزمن وخصوصاً على تطبيقات الزمن الحقيقي.

1.4.4.4.6. زمن التأخير من طرف إلى طرف في الشبكات ذات الطوبولوجيا المتشابكة:

في الشبكات ذات الطوبولوجيا المتشابكة انخفض أداء البروتوكول RIP حسب المحدد (التأخير من طرف إلى طرف) مقارنة بالبروتوكولين الآخرين، لتصل قيمة هذا المحدد إلى (1284 ms) للبروتوكول RIP و (1194.6 ms) للبروتوكول EIGRP بينما البروتوكول OSPF كانت قيمته الأفضل (1189.9 ms) وبفارق (94 ms) عن RIP.

2.4.4.4.6. زمن التأخير من طرف إلى طرف في الشبكات ذات الطوبولوجيا الحلقية:

وعلى النقيض من الشبكات ذات الطوبولوجيا المتشابكة، كان أداء البروتوكول RIP في الشبكات الحلقية مميزاً في هذا المحدد ولكن في الشبكات الصغيرة منها، حيث تفوق على البروتوكولين الآخرين بفارق وصل إلى (1266 ms) عن OSPF وليصل هذا الفارق إلى (1307 ms) عن البروتوكول EIGRP.

كما حصل البروتوكول EIGRP على قيم أفضل من البروتوكولين الآخرين في عدة شبكات متفرقة من كل نوع من أنواع الشبكات المدروسة، إلا أن البروتوكول OSPF كان بشكل عام هو الأفضل من حيث هذا المحدد في باقي الشبكات.

5.4.4.6. تحليل أداء الشبكات مع تغير الزمن:

من أجل توسيع عملية التحليل بشكل أكبر، تمت دراسة أداء الشبكات مع تغير الزمن عبر أخذ شبكة من الشبكات السابقة ومن ثم دراسة الإنتاجية والتأخير ضمنها وفق أزمنة تحليل مختلفة وذلك لمعرفة تغير قيم الإنتاجية وأزمنة

التأخير في الشبكات مع تغير الزمن، وأيضاً لمعرفة البروتوكول ذو الأداء الأفضل مع هذين المحددين ضمن الشبكة المدروسة، وكما هو معلوم فللحصول على أداء جيد في الشبكات، يجب تحقيق تأخير منخفض وإنتاجية عالية، حيث تُعرّف الإنتاجية على أنها أكبر قدر من حركة المرور المقبولة لنمط حركة مرور معينة عبر الشبكة. ويمكن تعريفها أيضاً على أنها مقدار المعلومات التي يتم تسليمها لكل وحدة زمنية بواسطة الشبكة. (Moudi M. *et al.*2020) [26].

لقد تمت دراسة شبكة ذات طوبولوجيا نجمية عن طريق القيام بتحليلها ثلاث مرات (مرة عند زمن محاكاة (10) دقائق ومرة ثانية عند زمن (20) دقيقة وأخيراً عند زمن محاكاة (30) دقيقة) لكل بروتوكول من بروتوكولات التوجيه السابقة وعند حمل فيديو منخفض بواسطة قياس المحددات: (تباين التأخير وانتاجية الشبكة والتأخير من طرف لطرف)، فحصلنا على النتائج التالية:

الجدول (6-3)- قيم المحددات للبروتوكول RIP مع تغير زمن المحاكاة عند نقل الفيديو

Simulation period (min)	10	20	30
Packet Delay Variation (ms)	29.515	133.97	197.78
End-to-End Delay (ms)	599.3	1188.5	1480.7
throughput (packets/s)	3946.1	4388.1	4525.5

الجدول (6-4)- قيم المحددات للبروتوكول OSPF مع تغير زمن المحاكاة عند نقل الفيديو

Simulation period (min)	10	20	30
Packet Delay Variation (ms)	29.101	132.70	189.51
End-to-End Delay (ms)	596.1	1183	1451.2
throughput (packets/s)	3948.7	4389.4	4544.3

الجدول (6-5)- قيم المحددات للبروتوكول EIGRP مع تغير زمن المحاكاة عند نقل الفيديو

Simulation period (min)	10	20	30
Packet Delay Variation (ms)	30.445	137.29	199.04
End-to-End Delay (ms)	609.6	1208.8	1496.6
throughput (packets/s)	3947.1	4388.7	4530.4

من دراسة الجداول السابقة نلاحظ ارتفاع قيم المحددات قيد الدراسة (تباين التأخير، التأخير من طرف إلى طرف، الإنتاجية) مع زيادة زمن المحاكاة للبروتوكولات الثلاث، ففي البروتوكول RIP كانت قيمة زمن تباين التأخير لهذا البروتوكول عند زمن محاكاة (10) دقائق تساوي (29.515 ms) لتزداد بشكل كبير عند المحاكاة لمدة (20) دقيقة فتصبح (133.97 ms)، وعند زيادة زمن المحاكاة إلى (30) دقيقة ارتفعت القيمة أكثر لتصل إلى (197.78 ms)، أما التأخير من طرف إلى طرف فكانت قيمته (599.3 ms) عند زمن محاكاة (10) دقائق وأصبحت بعد زيادة الزمن إلى (20) دقيقة (1188.5 ms) ثم لتصبح (1480.7 ms) عند زمن محاكاة (30) دقيقة، وبالنسبة للإنتاجية فكانت (3946.1 packets/s) عند زمن (10) دقائق وأصبحت بعد زيادة زمن المحاكاة لـ (20) دقيقة (4388.1 packets/s) ومن ثم ارتفعت إلى القيمة (4525.5 packets/s) بعد زيادة زمن المحاكاة إلى (30) دقيقة، أي أن أداء البروتوكول RIP أصبح أفضل في إنتاجية الشبكة فقط وتراجع في المحددين الآخرين مع زيادة الزمن.

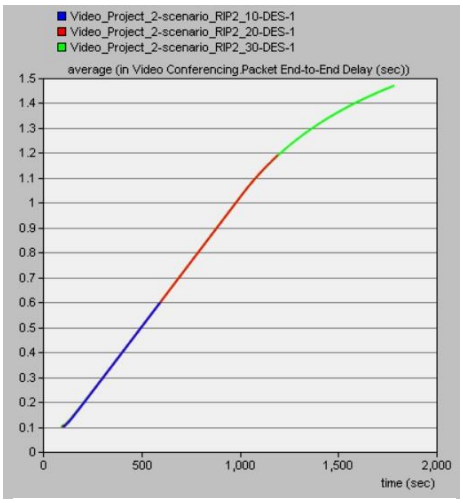
أما فيما يخص المحددات السابقة بالنسبة لبروتوكول التوجيه OSPF، فقد كانت قيمة زمن تباين التأخير تساوي (29.101 ms) عند محاكاة (10) دقائق لتزيد هذه القيمة إلى (132.70 ms) عند زمن محاكاة (20) دقيقة، ولتصبح (189.51 ms) عند زمن (30) دقيقة، وكانت قيمة زمن التأخير من طرف إلى طرف لها القيمة (596.1 ms) عند محاكاة مدتها (10) دقائق لتصبح (1183 ms) عند زمن (20) دقيقة ومن ثم لتصبح (1451.2 ms) عند زمن (30) دقيقة، فيما كانت قيمة الإنتاجية تساوي (3948.7 packets/s) لترتفع هذه القيمة إلى (4389.4 packets/s) عند زيادة زمن المحاكاة إلى (20) دقيقة ثم لتصبح (4544.3 packets/s) عند زمن محاكاة (30) دقيقة.

وأخيراً، ومن أجل قيم محددات البروتوكول EIGRP فقد كانت قيمة زمن تباين التأخير عند محاكاة لمدة (10) دقائق (30.445 ms) لتصبح عند زمن محاكاة مدته (20) دقيقة (137.29 ms) ثم لتزيد هذه القيمة إلى (199.04 ms) بعد زيادة زمن المحاكاة إلى (30) دقيقة، ومن أجل زمن التأخير من طرف إلى طرف كانت القيمة عند زمن محاكاة (10) دقائق (609.6 ms) ثم أصبحت (1208.8 ms) عند زمن (20) دقيقة ولتصبح عند زمن محاكاة (30) دقيقة (1496.6 ms)، وبالنسبة للإنتاجية، فكانت قيمتها (3947.1 packets/s) عند الزمن (10) دقائق لترتفع هذه القيمة

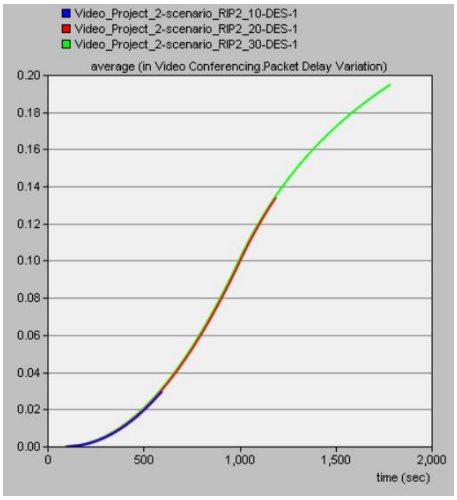
إلى (4388.7 packets/s) عند زمن محاكاة (20) دقيقة ولتصبح (4530.4 packets/s) عند زمن المحاكاة (30) دقيقة، أي أنه لم يتحسن أداء البروتوكولات الثلاث سوى في محدد الإنتاجية مع زيادة الزمن.

وباستخراج المخططات البيانية التي توضح القيم المتوسطة لهذه المحددات بالنسبة للزمن للبروتوكولات الثلاث المدروسة نجد ما يلي:

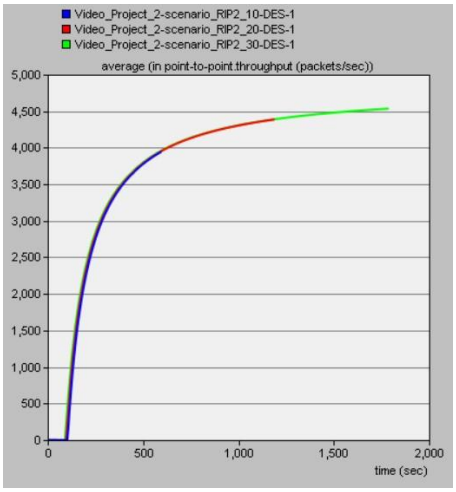
متوسط قيم محددات البروتوكول RIP:



الشكل (6-12)- زمن التأخير من طرف لطرف عند نقل الفيديو (سيناريو RIP)

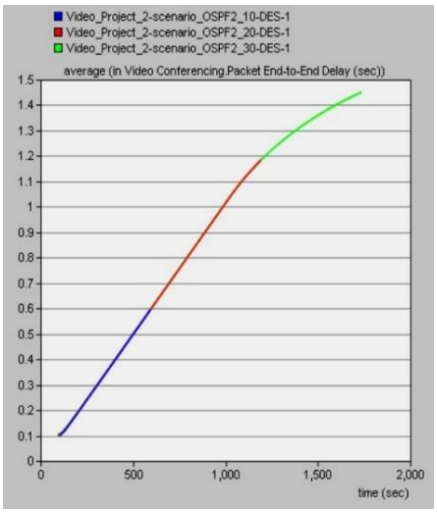


الشكل (6-13)- تباين التأخير عند نقل الفيديو (سيناريو RIP)

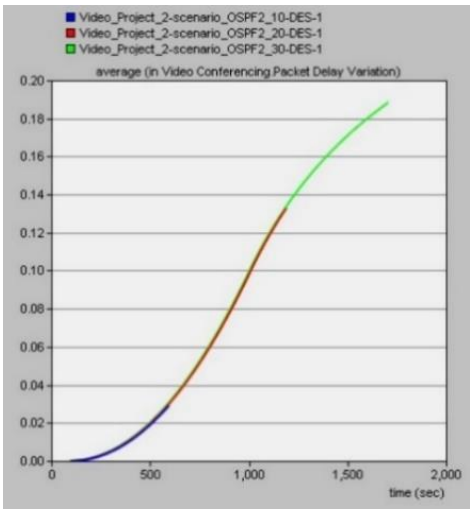


الشكل (6-14)- الإنتاجية عند نقل الفيديو (سيناريو RIP)

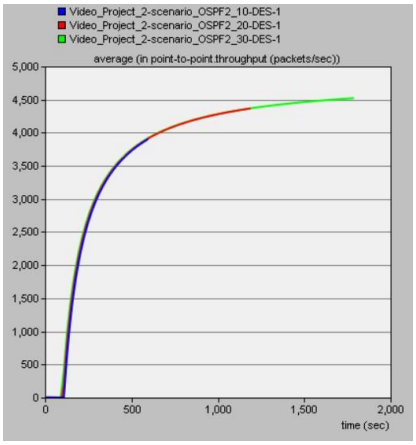
متوسط قيم محددات البروتوكول OSPF:



الشكل (6-15)- زمن التأخير من طرف لطرف عند نقل الفيديو (سيناريو OSPF)

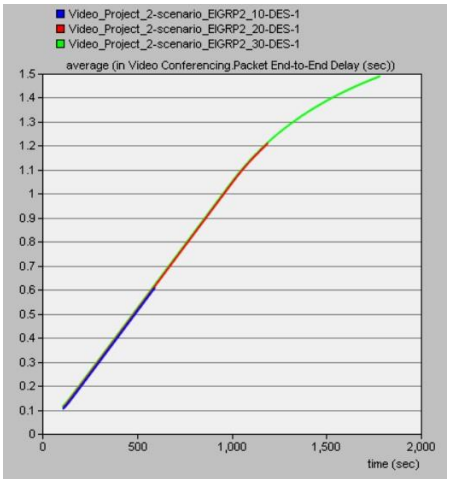


الشكل (6-16)- تباين التأخير عند نقل الفيديو (سيناريو OSPF)

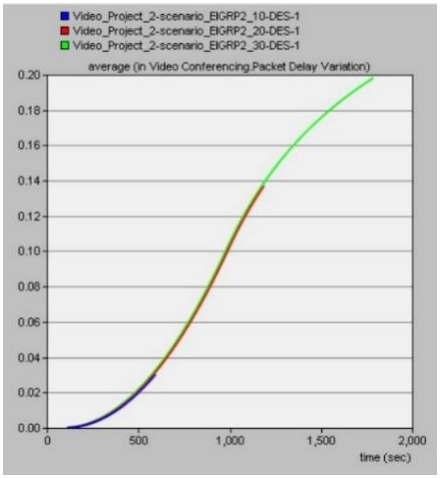


الشكل (6-17)- الإنتاجية عند نقل الفيديو (سيناريو OSPF)

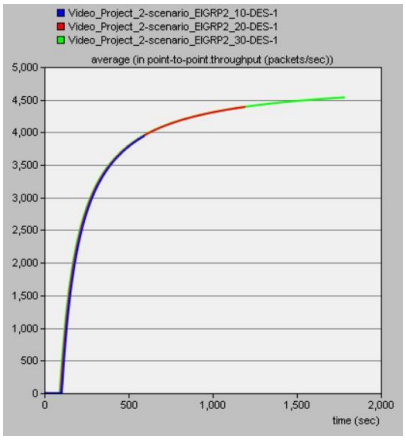
متوسط قيم محددات البروتوكول EIGRP:



الشكل (6-18)- زمن التأخير من طرف لطرف عند نقل الفيديو (سيناريو EIGRP)



الشكل (6-19)- تباين التأخير عند نقل الفيديو (سيناريو EIGRP)



الشكل (6–20)– الإنتاجية عند نقل الفيديو (سيناريو EIGRP)

من دراسة الأشكال البيانية السابقة والجداول الخاصة بها نستنتج ارتفاع قيم التأخير بزيادة الزمن بشكل كبير، في حين ازدادت أيضاً قيمة الإنتاجية ولكن بشكل أقل وخصوصاً بعد زمن معين، فكما ذكرنا سابقاً أن التأخيرات يمكن أن تتراكم وتؤثر بشكل كبير على أداء الشبكة مع مرور الزمن وتحديداً على تطبيقات الزمن الحقيقي، لذلك كان لابد من الأخذ بعين الاعتبار هذه التأخيرات عند تصميم الشبكة لما له دور كبير على أدائها.

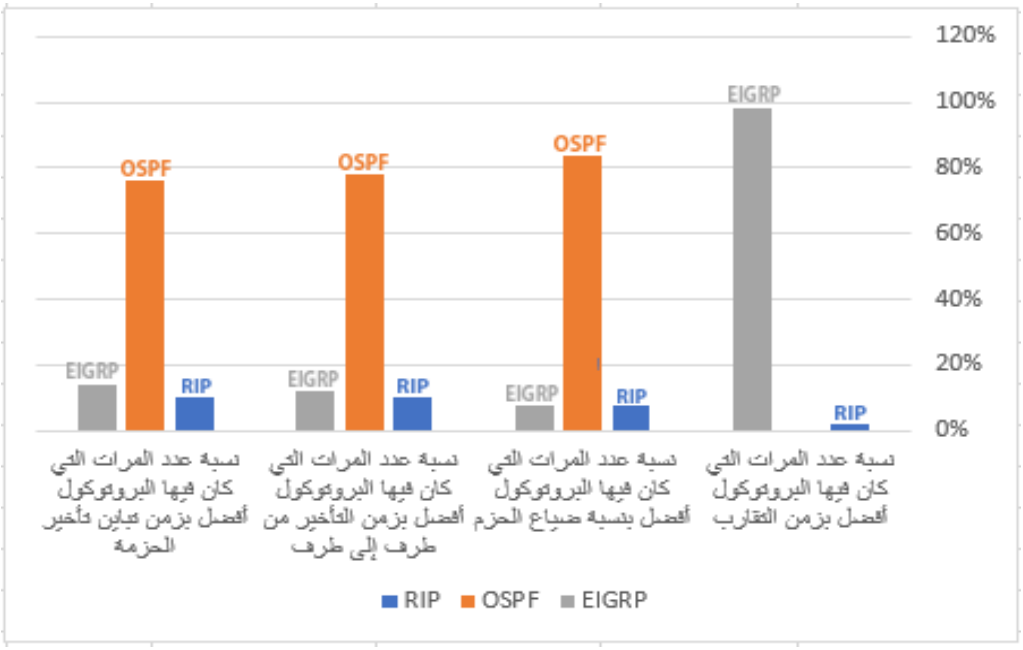
وبالمقارنة بين أداء بروتوكولات التوجيه حسب نتائج الدراسة التي حصلنا عليها، وجدنا زيادة الفرق بين قيم البروتوكولات مع زيادة الزمن، ووجدنا أن البروتوكول OSPF كانت له بشكل عام قيم أفضل من البروتوكولين الآخرين في محددات الدراسة (تباين التأخير، التأخير من طرف إلى طرف، الإنتاجية).

وأخيراً، وللوصول إلى نتيجة عامة من هذه الدراسة من أجل تحديد البروتوكول الأفضل وفق المحددات التي ذكرناها في بداية هذا البحث، فإنه تم إيجاد نسبة عدد المرات التي تفوّق بها كل بروتوكول من البروتوكولات السابقة على نظيريه في كل محدد من المحددات الرئيسية، وذلك بحاصل قسمة عدد المرات التي كان فيها البروتوكول أفضل بالأداء من البروتوكولات الأخرى على عدد الشبكات الكلي (50 شبكة)، كما هو موضح بالجدول (6–6):

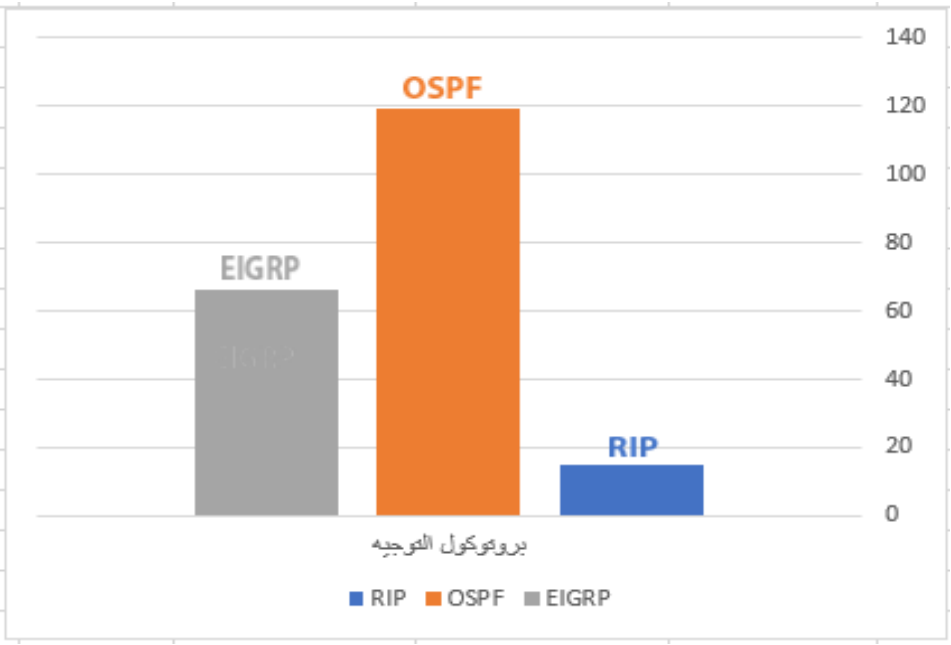
الجدول (6–6)– نسبة عدد المرات التي تفوّق بها كل بروتوكول حسب محددات الدراسة عند نقل الفيديو

البروتوكول	نسبة عدد المرات التي كان فيها البروتوكول أفضل بزمن تباين تأخير الحزمة	نسبة عدد المرات التي كان فيها البروتوكول أفضل بزمن التأخير من طرف إلى طرف	نسبة عدد المرات التي كان فيها البروتوكول أفضل بنسبة ضياع الحزم	نسبة عدد المرات التي كان فيها البروتوكول أفضل بزمن التقارب
RIP	10%	10%	8%	2%
OSPF	76%	78%	84%	0%
EIGRP	14%	12%	8%	98%

ويمكن تمثيل الجدول (6-6) بيانياً كما بالشكلين (6-21) و (6-22):



الشكل (6-21)- نسبة عدد الشبكات التي حققت فيها بروتوكولات التوجيه أفضل أداء لكل محدد من محددات الدراسة عند نقل الفيديو



الشكل (6-22)- عدد المرات الكلي التي حقق فيها بروتوكول التوجيه أفضل أداء بالنسبة لمحددات الدراسة عند نقل الفيديو

5.4.6. الخلاصة:

بدراسة النتائج الموضحة بالجدول (6-6) والشكلين (6-21) و (6-22) نلاحظ أن البروتوكول OSPF قد تفوق بعدد المرات التي أحرز فيها أفضل أداء وذلك بمحددات الدراسة كلها ما عدا المحدد (زمن تقارب البروتوكول) حيث كان الأداء الأفضل في هذا المحدد من نصيب البروتوكول EIGRP وبفارق كبير عن باقي البروتوكولات في الشبكات المدروسة.

فقد حقق البروتوكول OSPF أداءً أفضل من البروتوكولين الآخرين في المحدد (تباين تأخير حزم الفيديو) حيث حصل على نسبة عدد المرات التي كان فيها أفضل من البروتوكولين الآخرين وهي 76% من عدد الشبكات المدروسة فيما حصل البروتوكولين الآخرين على نسب أقل بشكل ملحوظ مقارنة معه، فحصل البروتوكول EIGRP على 14% أما البروتوكول RIP فقد حصل على 10% من عدد الشبكات المدروسة.

أما فيما يخص المحدد الثاني وهو (زمن تأخير حزم الفيديو من طرف إلى طرف) فكان الأداء الأفضل في هذا المحدد أيضاً من صالح البروتوكول OSPF حيث حصل على نسبة 78%، أما البروتوكولين الآخرين فحصلا على نسب أقل بكثير مقارنة معه، فنال البروتوكول EIGRP على نسبة 16%، أما البروتوكول RIP فحصل على 10% من عدد الشبكات التي كان فيها أفضل من البروتوكولين الآخرين بالنسبة للشبكات المدروسة.

وفي المحدد الثالث (نسبة ضياع حزم الفيديو) نجد أيضاً الفارق الكبير بين نتائج البروتوكول OSPF والبروتوكولين الآخرين، فقد نال البروتوكول OSPF المرتبة الأولى وبنسبة 84% وهي نسبة عدد المرات التي كان فيها هذا البروتوكول أفضل من البروتوكولين الآخرين واللذين حصلا على نسبتين متساويتين وهي 8% من عدد الشبكات المدروسة.

أما بالمحدد الأخير وهو (زمن تقارب البروتوكول) فكان الأداء الأفضل للبروتوكول EIGRP في هذه المرة وبنسبة 98%، أما البروتوكول OSPF فلم ينافس EIGRP الذي كان زمن تقاربه ضمن الشبكات المدروسة أقل بكثير منه، فيما حصل البروتوكول RIP على النسبة 2% من عدد الشبكات المدروسة.

وبإحصاء عدد المرات التي حقق فيها كل بروتوكول أفضل النتائج بالنسبة لمحددات الدراسة جميعها، نجد أن البروتوكول OSPF قد حقق أفضل نتيجة وبمجموع (119) مرة من أصل 200 (50 شبكة x 4 محددات دراسة لكل شبكة)، في حين أتى البروتوكول EIGRP في المرتبة الثانية وبفارق كبير نسبياً وبمجموع يساوي (66) مرة، أما البروتوكول RIP فقد كان الأخير في الأداء وبمجموع (15) مرة بالشبكات المدروسة وحسب محددات هذه الدراسة عند نقل خدمة الفيديو وذلك حسب الشكل (6-22).

لذلك نجد أن الأداء الأفضل وبشكل عام ضمن هذه العينة من الشبكات المدروسة عند نقل خدمة الفيديو وحسب محددات هذه الدراسة كان من نصيب البروتوكول OSPF.

نستنتج من هذه الدراسة أن أداء بروتوكول التوجيه OSPF عند تغير بعض العوامل مثل حجم الشبكة، وشكل الطوبولوجيا الخاصة بها، وحجم البيانات المرسلة ضمنها عند نقل خدمة الفيديو، كان أعلى أداء حسب محددات الدراسة ضمن الشبكات المدروسة.

5.6. دراسة تأثير بروتوكول التوجيه عند نقل الصوت:

1.5.6. محددات القياس ضمن الدراسة:

في سياق هذه الدراسة، تم تحديد عدة محددات لقياس أداء البروتوكولات، من بينها:

- تذبذب الصوت (Jitter)
- قيمة متوسط نقاط الرأي للصوت (MOS)
- زمن تأخير الإيثرنت
- زمن تأخير حزم الصوت من طرف إلى طرف
- تباين تأخير حزم الصوت

حيث يقوم تحليل هذه المعايير بدور هام في تفسير أداء البروتوكولات ضمن سيناريوهات متنوعة، وفيما يلي شرح بسيط لكل منها:

1.1.5.6. تذبذب الصوت (Jitter):

يشير محدد تذبذب الصوت إلى التفاوت في الفاصل الزمني بين حزم البيانات المتعاقبة، ويحدث ذلك نتيجة لازدحام الشبكة، أو ضعف الأجهزة، أو عدم تحديد أولويات لتسليم الحزم. يمكن للتذبذب أن يؤثر سلباً على جودة التطبيقات التي تعتمد على الزمن الحقيقي مثل VoIP، حيث يعتبر انحرافاً عن دورية التكرار المتوقع للإشارة، ويُعتبر مؤشراً أساسياً لأداء الأنظمة المتصلة بالشبكة.

2.1.5.6. قيمة متوسط نقاط الرأي للصوت (MOS: Mean Opinion Score):

هو مقياس لجودة الصوت تم استخدامه في الاتصالات الهاتفية لعقود من الزمن لتقييم آراء المستخدمين حول جودة المكالمات. يُستخدم هذا الاختبار على نطاق واسع في شبكات نقل الصوت عبر بروتوكول الإنترنت (VoIP) لضمان جودة نقل الصوت واختبار مشكلات الجودة وتوفير مقياس يمكن من خلاله قياس تدهور الصوت والأداء. يعد تسجيل MOS أمراً ضرورياً لضمان رضى المستخدمين. يعين القياس درجة من 1 إلى 5 للإشارة إلى جودة مكالمة VoIP بعد إرسالها وضغطها باستخدام برامج الترميز.

3.1.5.6. زمن تأخير الإيثرنت:

هو مقدار الوقت الذي تستغرقه حزمة من البيانات الصوتية للانتقال عبر أجهزة متعددة، ثم يتم استلامها في وجهتها. عندما تكون فترات التأخير في الإرسال صغيرة، يُشار إليها على أنها شبكة ذات زمن وصول منخفض (مرغوب فيه وخاصة بتطبيقات الزمن الحقيقي) ويُشار إلى التأخيرات الأطول بشبكة ذات زمن وصول مرتفع (غير مرغوب فيه كثيراً). عادة ما يقوم المهندسون بالإبلاغ عن كل من التأخير الأقصى والمتوسط، مع ملاحظة أن التأخير يقسم إلى عدة أجزاء: (تأخير المعالجة، وتأخير الانتظار، وتأخير الإرسال).

4.1.5.6. زمن تأخير حزم الصوت من طرف إلى طرف:

يشير التأخير من طرف إلى طرف أو التأخير في اتجاه واحد إلى الوقت المستغرق لإرسال الحزمة عبر الشبكة من المصدر إلى الوجهة. وهو مجموع التأخيرات التي حدثت في سلسلة من العقد الوسيطة في الطريق إلى الوجهة مع الأخذ بعين الاعتبار الانتظار وتأخر نشر الحزم.

5.1.5.6. تباين تأخير حزم الصوت:

وهو التباين بين التأخيرات من طرف إلى طرف لحزم الصوت، حيث يقاس التأخير من طرف إلى طرف للحزمة من وقت إنشائها إلى وقت استقبالها، ويعتبر هذا القياس أمراً حيوياً عند نقل الصوت، نظراً لتأثيره في فقدان الحزم وعدم قدرتها على الاسترجاع.

2.5.6. نتائج التحليل:

بعد إجراء تحليل الشبكات التي تمت دراستها وفقاً للمحددات المذكورة (تذبذب الصوت (Jitter)، قيمة متوسط نقاط الرأي للصوت (MOS)، زمن تأخير الإيثرنت، تأخير حزم الصوت من طرف إلى طرف، تباين تأخير حزم الصوت) وذلك عند تغير العوامل التي تؤثر فيها (حجم الشبكة، شكل طوبولوجيا الشبكة، كمية البيانات في الشبكة)، حيث تدرج حجم الحزم المرسلة بين 512–1500 بايت، كانت المحددات متباينة بالقيم حيث تراوحت قيم تذبذب الصوت بين (28 و 248) µs، وقيم متوسط نقاط الرأي للصوت بين (2.6 و 3.96)، وزمن تأخير الإيثرنت بين (60 و 278) µs، أما تأخير حزم الصوت من طرف إلى طرف فتراوحت القيم بين (120 و 418) ms، وأخيراً تباين تأخير حزم الصوت فكانت القيم تتراوح بين (218 , 20) µs .

تم إيجاد نسبة عدد المرات التي تفوّق بها كل بروتوكول من البروتوكولات السابقة على نظيريه في كل محدد من محددات هذه الدراسة (بتقسيم عدد المرات التي كان فيها أداؤه أفضل من البروتوكولين الآخرين على عدد الشبكات المدروسة) كما هو موضح بالجدول (6–7):

الجدول (6–7)– نسبة عدد المرات التي تفوّق فيها كل بروتوكول توجيه حسب محددات الدراسة عند نقل الصوت

	نسبة عدد المرات التي كان فيها البروتوكول أفضل بتذبذب الصوت (Jitter)	نسبة عدد المرات التي كان فيها البروتوكول أفضل بقيمة متوسط نقاط الرأي للصوت (MOS)	نسبة عدد المرات التي كان فيها البروتوكول أفضل بتباين تأخير حزم الصوت	نسبة عدد المرات التي كان فيها البروتوكول أفضل بزمن تأخير حزم الصوت من طرف إلى طرف	نسبة عدد المرات التي كان فيها البروتوكول أفضل بزمن تأخير الإيثرنت
RIP	32%	70%	24%	44%	14%
OSPF	40%	74%	72%	46%	28%
EIGRP	28%	70%	82%	56%	62%

ويمكن تمثيل الجدول (6–7) بيانياً كما هو موضح بالشكلين (6–23) و (6–24):



الشكل (6–23)– نسبة عدد الشبكات التي حققت فيها بروتوكولات التوجيه أفضل أداء لكل محدد من محددات الدراسة عند نقل الصوت



الشكل (6-24)- عدد المرات الكلي التي حقق فيها بروتوكول التوجيه أفضل أداء بالنسبة لمحددات الدراسة عند نقل الصوت

3.5.6. الاستنتاجات:

تبين لدينا من النتائج السابقة أن البروتوكول OSPF حاز على أعلى نسبة كان فيها أفضل أداءً من البروتوكولين الآخرين عند قياس تذبذب الصوت (Jitter) والتي كانت 40% فيما كانت نسبة البروتوكولين RIP و EIGRP تساوي 32% و 28% على التوالي، وحصل أيضاً على أعلى نسبة عند قياس قيمة متوسط نقاط الرأي (MOS) ولكن بعدد مرات متقاربة من البروتوكولين الآخرين حيث حصل على نسبة 74% بينما نال البروتوكولان الآخران 70%، وأما فيما يتعلق ببقية المحددات فقد حصل البروتوكول EIGRP على أعلى نسبة كان فيها أفضل من البروتوكولين الآخرين عند بقية المحددات التي تم دراستها (زمن تأخير الإيثرنت، زمن تأخير حزم الصوت من طرف إلى طرف، تباين تأخير حزم الصوت) حيث حصل في المحدد الأول على نسبة 82% بينما حصل RIP على 24% أما OSPF فكان 72%، أما عند قياس المحدد الثاني فقد حصل EIGRP على نسبة 56% بينما نال RIP و OSPF على نسب أقل وتساوي 44% و 46% على التوالي، وأيضاً حصل EIGRP على أعلى نسب في المحدد الأخير حيث نال نسبة 62% بينما حصل RIP على نسبة 14% وحصل OSPF على النسبة 28%.

لذلك نستنتج أن البروتوكول EIGRP قد حصل على أفضل نسب عدد من المرات تقريباً التي كان فيها أفضل من حيث الأداء من البروتوكولات الأخرى عند محددات الدراسة السابقة.

الفصل السابع

إنشاء وتطوير الخوارزمية المقترحة

Create and develop the proposed algorithm

1.7. مقدمة عن الخوارزمية المقترحة

من المعروف أن شبكات الحاسوب الحديثة تشهد تنوّعاً كبيراً في أنماط حركة المرور (Traffic) ومتطلّباتها الزمنية وجودتها من فيديو فوري عالي الحساسية للتذبذب (Jitter) والتأخير (Delay)، إلى قواعد بيانات تحتاج موثوقية وسرعة تقارب في الزمن الحقيقي، مروراً بالصوت الذي يتطلّب انتظاماً واستقراراً. ورغم نضج بروتوكولات التوجيه التقليدية مثل OSPF و EIGRP، فإنها تُطبَّق عادةً على الشبكة ككلّ دون مراعاة دقيقة لاختلاف خصائص البيانات ونوعيّتها في لحظةٍ بعينها. من هنا تبرز الحاجة إلى منهجٍ انتقائيّ ديناميكي يختار بروتوكول التوجيه الأنسب حسب نوع البيانات وظروف الشبكة، بدلاً من فرض بروتوكول واحدٍ بصورةٍ جامدة.

في هذا القسم من البحث طورنا خوارزمية مستوحاة من مستعمرة النمل (Ant Colony) مهمّتها تعيين بروتوكول التوجيه الملائم لكل مسار كامل بين المصدر والهدف، اعتماداً على الفيرمون الخاص بكل بروتوكول. لا تهدف الخوارزمية إلى إيجاد " أقصر" طريق كما بالخوارزمية التقليدية، بل إلى تصنيف المسارات وإسنادها إلى أحد البروتوكولين OSPF أو EIGRP (وذلك استناداً إلى النتائج التي حصلنا عليها بالفصل الخامس) حسب نوع البيانات (فيدو، صوت، نصوص)، بحيث تعمل بروتوكولات التوجيه لاحقاً ضمن مسارات ملوّنة ومحدّدة مسبقاً بما يتوافق مع طبيعة الحمل.

2.7. آلية عمل الخوارزمية المقترحة

- سيتم في هذه الخوارزمية تحديد بروتوكول التوجيه المناسب لكل شبكة بشكل آلي حسب نوع البيانات التي يتم إرسالها ضمنها، حيث يتم فحص كل وصلة من المصدر إلى الهدف ومن ثم يتم تطبيق بروتوكول التوجيه المناسب لها وذلك باستخدام تعديل خوارزمية مستعمرة النمل (ACO).
- كل نوع بيانات له فيرمون محدد يتم قياسه بشكل متكرر لمعرفة نوع البيانات الذي يتم إرساله عن طريق النمل الاصطناعي (حزم بيانات صغيرة جداً).

- بناءً على نوع الفيرمون، يتم تشغيل بروتوكول التوجيه المناسب للشبكة.
- لمعرفة بروتوكول التوجيه المناسب لكل نوع من أنواع البيانات تم دراسة الشبكات في القسم الأول.

3.7. الخوارزمية المقترحة

إن توضيح آلية عمل الخوارزمية المقترحة يتم باتخاذ الخطوات التالية:

تهيئة المحددات الأولية للخوارزمية:

- تحديد عدد النمل في المستعمرة.
- تهيئة مستويات الفيرمون لكل بروتوكول توجيه (OSPF، EIGRP) على كل وصلة من قفزة إلى قفزة إلى قيمة موجبة صغيرة.
- حدد بروتوكول التوجيه EIGRP إذا كان نوع البيانات ملفات نصية.
- حدد بروتوكول التوجيه OSPF إذا كان نوع البيانات فيديو.
- حدد بروتوكول التوجيه EIGRP إذا كان نوع البيانات صوت.

إنشاء النمل والوصول للوجهة وفق مراحل دورة حياة النمل بالخوارزمية المطورة:

1- المرحلة الأولى من دورة حياة النمل:

- إنشاء نملة جديدة
- تبدأ كل نملة من عقدة المصدر.
- تسير النملة عشوائياً عبر الشبكة بحثاً عن عميل متوافق مع نوع البيانات (Video، Voice، Text).

- تستخرج النملة خلال رحلتها نوع البيانات من الحزم المارة في الشبكة.
- تستكشف النملة المسارات الممكنة مع القيود التالية:
  - يتم إزالة أي انحياز احتمالي في اختيار العقد (جعل جميع الخيارات متساوية الاحتمال).
  - يتم تعطيل تبخر الفيرمون، للحفاظ على جميع المعلومات المكتشفة.
  - لا يُسمح للنملة بزيارة نفس العقدة أكثر من مرتين (لمنع الحلقات).
  - لا يُسمح للنملة بالتوجه إلى عميل غير متوافق
- التحقق من صحة الهدف
- تحفظ النملة المسارات الكاملة فقط، ويُطبق حد أقصى لطول المسار (30 عقدة).
- يتم تصنيف كل مسار حسب نوع الخدمة (نص/صوت/فيديو).
- يتم استبعاد أي مسار يصل إلى عميل غير تابع لنوع الخدمة الصحيح.
- انتقال النملة إلى مرحلة العودة

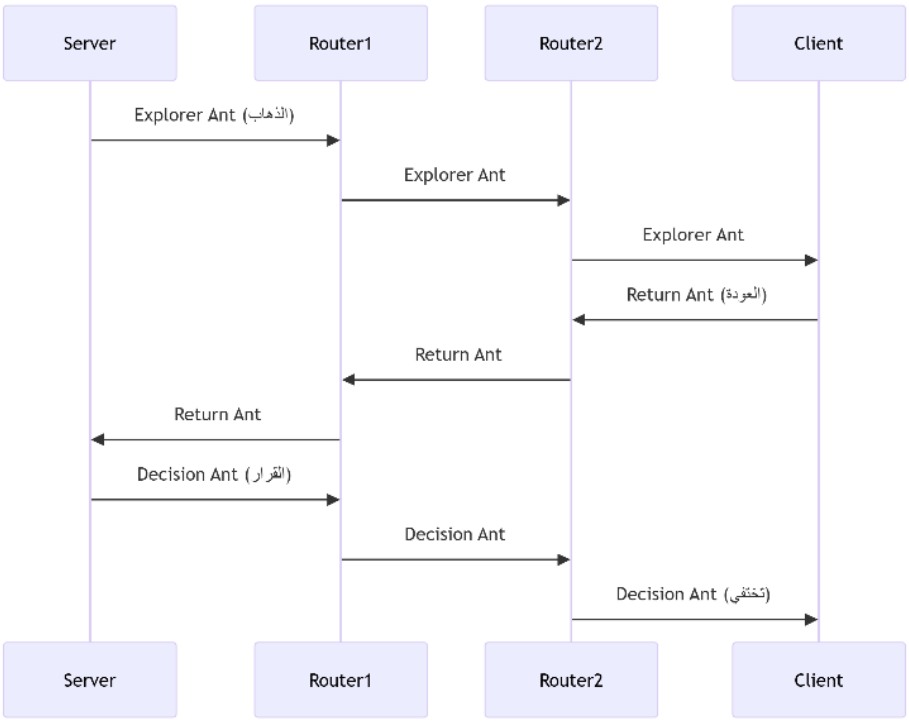
2- المرحلة الثانية من دورة حياة النمل:

- عند الوصول إلى العقدة الهدف (العميل الصحيح المتوافق مع نوع السيرفر)، تعود النملة إلى المصدر.
- أثناء العودة (مرحلة RETURN):
  - يتم تسجيل المسار الكامل إلى العميل.
  - يتم عكس المسار (reverse) لتبدأ العودة إلى المصدر.

- في طريق العودة، تستمر النملة بتسجيل الفيرمونات بناءً على نوع البيانات
- يتم تقييم المسار بالكامل (حسب نوع البروتوكول المناسب).
- حساب الفيرمون النهائي التراكمي للبروتوكولين (OSPF و EIGRP) على كامل المسار بناءً على هذا التقييم.

3- المرحلة الثالثة من دورة حياة النمل:

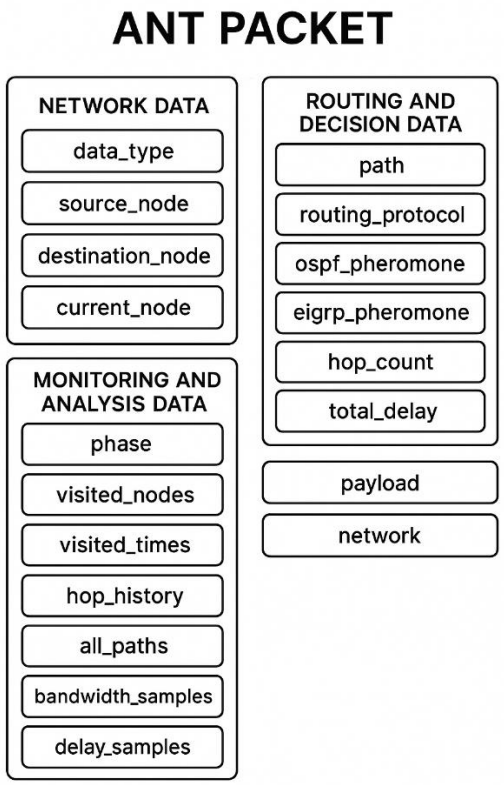
- النملة تنطلق من المصدر مرة أخرى، عبر نفس المسار، ولكن بهدف إخبار كل راوتر على المسار بالبروتوكول المختار.
- كل راوتر في هذا المسار يفعّل البروتوكول المناسب حسب قرار النملة.
- عند الوصول إلى العميل، تختفي النملة.



الشكل (7-1)- مراحل دورة حياة النمل بالخوارزمية المقترحة

4.7. حقول النملة المقترحة:

يوضح الشكل (7–2) حقول النملة بالخوارزمية المقترحة، فيما يوضح الجدول (7–1) الحقول ضمن الشكل.



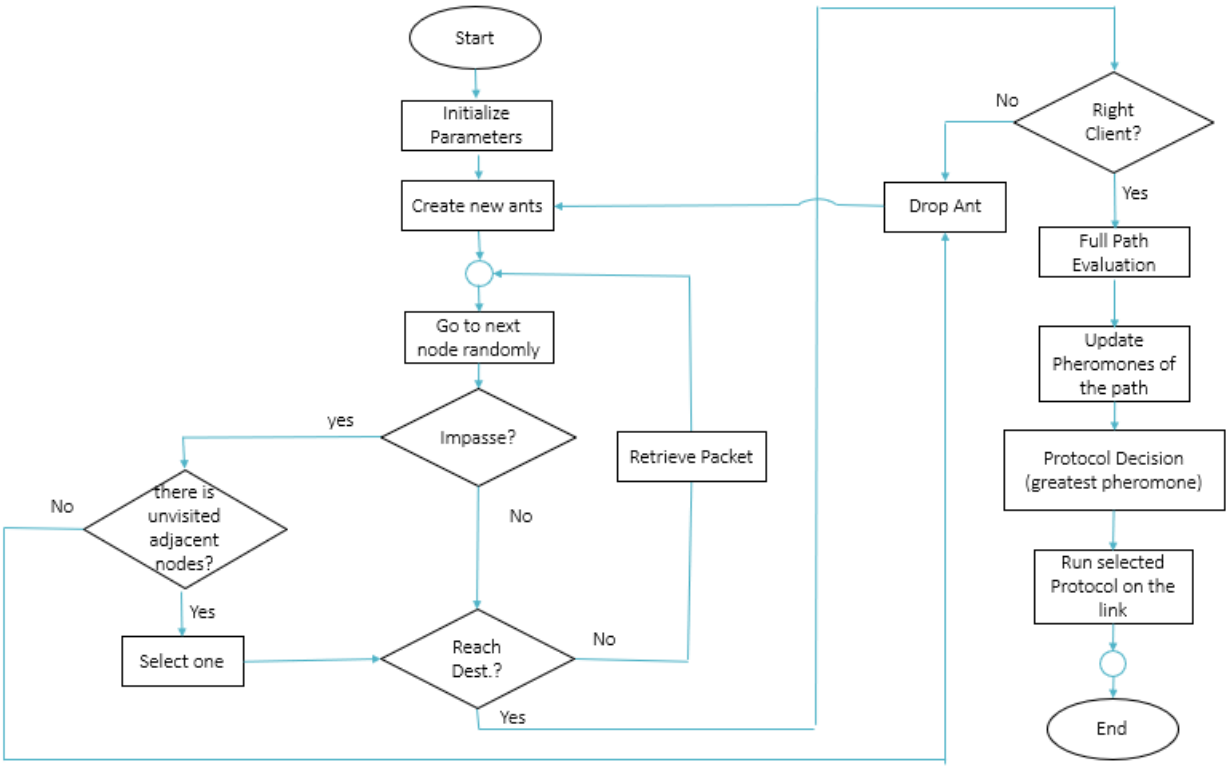
الشكل (7–2)– حقول النملة بالخوارزمية المقترحة

الجدول (7–1)– تفصيل حقول النملة بالخوارزمية المقترحة

الحقل	القيمة داخل الشبكة أثناء التشغيل
data_type	"video" أو "voice" أو "database" حسب نوع الحزمة
source_node	مثال "S1" :سيرفر فيديو
destination_node	مثال "C3" :عميل فيديو
current_node	يتغير عند كل قفزة، مثل "R2" ثم "R5" ثم "C3"
path	قائمة العقد التي مرت بها النملة، مثل ["S1", "R2", "R5", "C3"]رت بها النملة، مثل ["S1", "R2", "R5", "C3"]
routing_protocol	"OSPF" إذا كانت الحزمة فيديو، "EIGRP" إذا كانت صوت أو قواعد بيانات
ospf_pheromone	قيمة عددية مثل 0.78 تحسب عند العودة للمصدر

eigrp_pheromone	قيمة عددية مثل 0.65 تحسب عند العودة للمصدر
hop_count	عدد القفزات، مثل 3
total_delay	مجموع التأخير، مثل 52 ms
phase	"FORWARD" أو "RETURN" أو"DECISION"
visited_nodes	قائمة مثل ["R2", "R5"]
visited_times	عدد مرات زيارة كل عقدة، مثل {"R2": 1, "R5": 1}
hop_history	تسلسل القفزات، مثل ["S1→R2", "R2→R5", "R5→C3"]
all_paths	قائمة بكل المسارات المكتشفة، مثل: [["S1", "R2", "R5", "C3"], ["S1", "R3", "R6", "C3"]]
bandwidth_samples	قيم مثل [100 Mbps, 80 Mbps, 90 Mbps]
delay_samples	قيم مثل [20 ms, 15 ms, 17 ms]
payload	محتوى الحزمة، مثل "video_stream"
network	اسم الشبكة أو معرفها، مثل "Dev-ACO"

ويمكن تمثيل الخوارزمية المقترحة بالمخطط الموضح بالشكل (7-3):



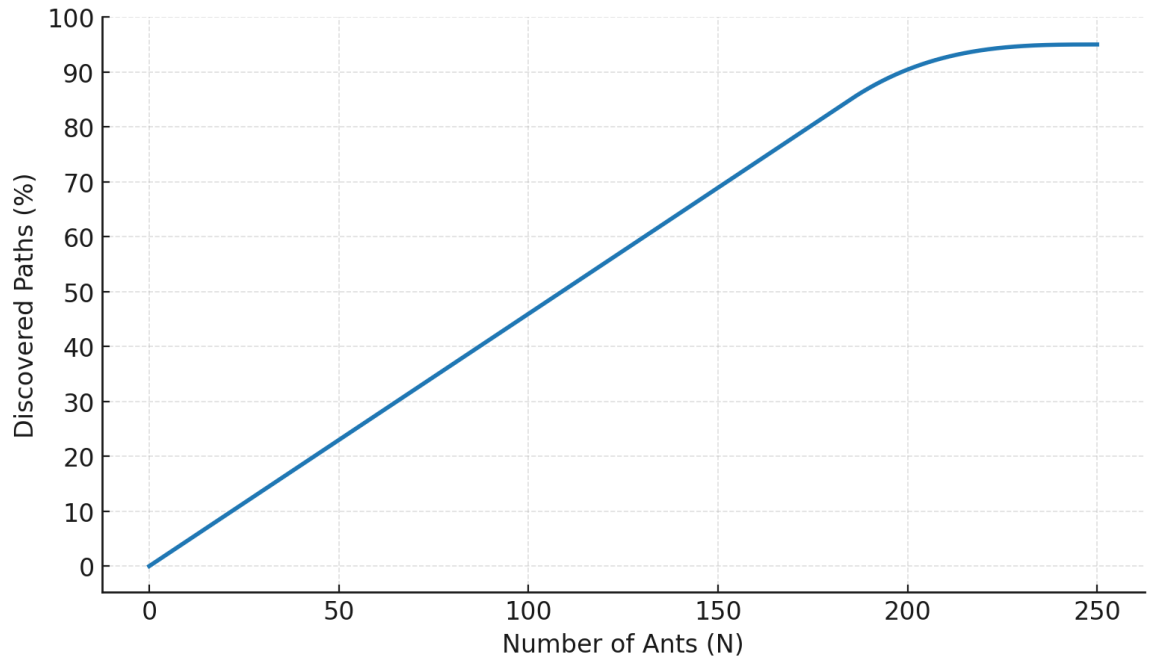
الشكل (7-3)- مخطط الخوارزمية المقترحة

5.7. أثر زيادة عدد النمل على اكتشاف المسارات:

يبيّن تحليل النتائج أن ازدياد عدد النمل المولَّد في مرحلة الاستكشاف يؤدي إلى ارتفاع تدريجي في نسبة المسارات المكتشفة ضمن الشبكة. ففي المراحل الأولى يكون معدل الاكتشاف مرتفعاً، إذ تساهم كل نملة جديدة في إضافة مسارات غير مكتشفة سابقاً، مما يعكس تزايداً سريعاً في التغطية الكلية. ومع التوسع في عدد النمل المولَّد، تبدأ نسبة الاكتشاف بالاقتراب من حدٍّ أعلى، حيث إن معظم المسارات الممكنة تكون قد تم التعرف عليها مسبقاً، وبالتالي تصبح مساهمة النمل الإضافي أقل تأثيراً. وعند بلوغ عدد معين من النمل، يُسجَّل ما يُعرف بحالة الإشباع، إذ يتم عندها اكتشاف جميع المسارات تقريباً، ويستقر منحني الأداء عند قيمة شبه ثابتة بحيث لا تحقق الزيادة في عدد النمل أي تحسّن إضافي ملحوظ في التغطية.

ومن الناحية العملية، تُظهر هذه النتائج العلاقة بين الكفاءة والتكلفة في تصميم الخوارزمية. فبينما تسهم الزيادة التدريجية في عدد النمل في تحسين التغطية خلال المراحل الأولى، إلا أنّ توليد أعداد أكبر بعد الوصول إلى حالة الإشباع لا يضيف قيمة فعلية من حيث اكتشاف مسارات جديدة. وبالتالي فإن الموارد الحسابية والزمنية المستهلكة في تشغيل نمل إضافي بعد هذه النقطة تُعدّ غير فعّالة. وعليه، يُوصى باختيار عدد النمل الأمثل الذي يضمن تحقيق تغطية شبه كاملة للمسارات مع تقليل الهدر في الموارد، الأمر الذي ينعكس إيجاباً على كفاءة الأداء الكلي للشبكة.

يوضح المخطط البياني بالشكل (7–4) علاقة زيادة عدد النمل مع اكتشاف مسارات جديدة ضمن الشبكة.



الشكل (7-4)- علاقة زيادة عدد النمل مع اكتشاف المسارات

6.7. نتائج المحاكاة:

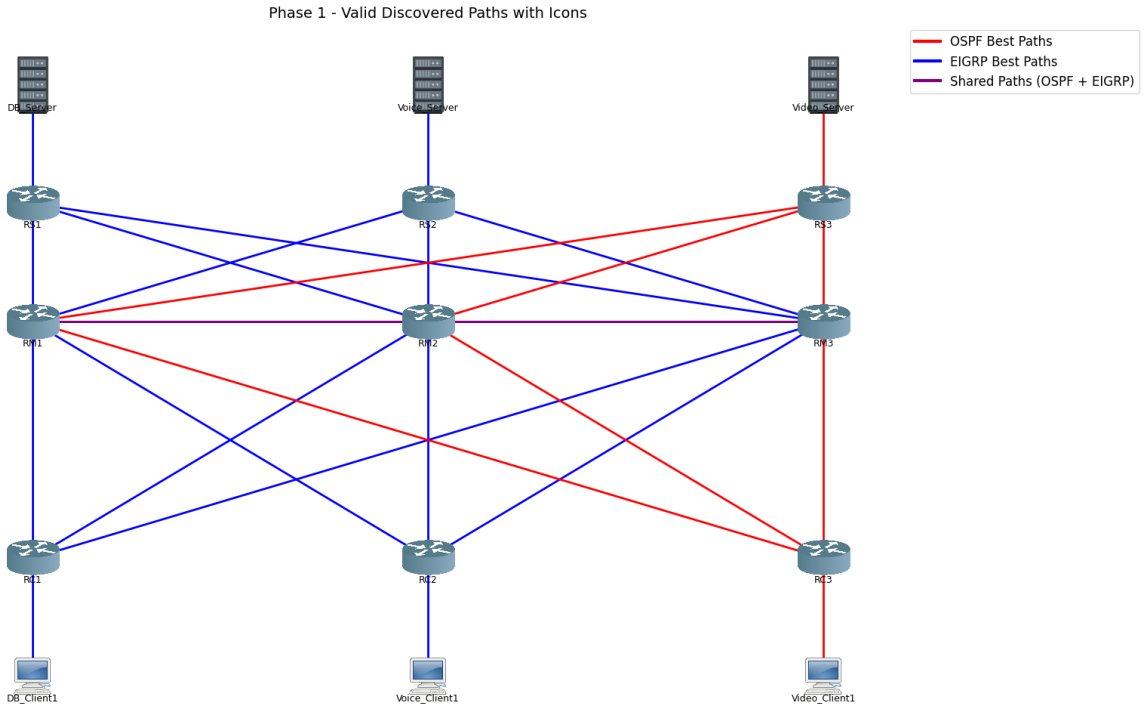
```
=== Phase 1: Path Discovery ===
Valid Ant Ant-DB_Server-DB_Client1
EIGRP pheromone += 0.50 → 1.50
EIGRP pheromone += 0.50 → 2.00
EIGRP pheromone += 0.50 → 2.50
EIGRP pheromone += 0.50 → 3.00
EIGRP pheromone += 0.50 → 3.50
EIGRP pheromone += 0.50 → 4.00
EIGRP pheromone += 0.50 → 4.50
EIGRP pheromone += 0.50 → 5.00
EIGRP pheromone += 0.50 → 5.50
EIGRP pheromone += 0.50 → 6.00
EIGRP pheromone += 0.50 → 6.50
EIGRP pheromone += 0.50 → 7.00
EIGRP pheromone += 0.50 → 7.50
EIGRP pheromone += 0.50 → 8.00
EIGRP pheromone += 0.50 → 8.50
EIGRP pheromone += 0.50 → 9.00

Ant Ant-DB_Server-DB_Client1-1755093345.4554975 reached valid destination
Ant Ant-DB_Server-DB_Client1-1755093345.4554975 returned to source
Pheromones retained | EIGRP: 8.00, OSPF: 0.50
Protocol → EIGRP | EIGRP: 8.00 | OSPF: 0.50
Router RS1 set to EIGRP
Router RM3 set to EIGRP
Router RS3 set to EIGRP
Router RM2 set to EIGRP
Router RC3 set to EIGRP
Router RM1 set to EIGRP
Router RC1 set to EIGRP
Router RS2 set to EIGRP
Ant Ant-DB_Server-DB_Client1-1755093345.4554975 completed decision phase and reached destination

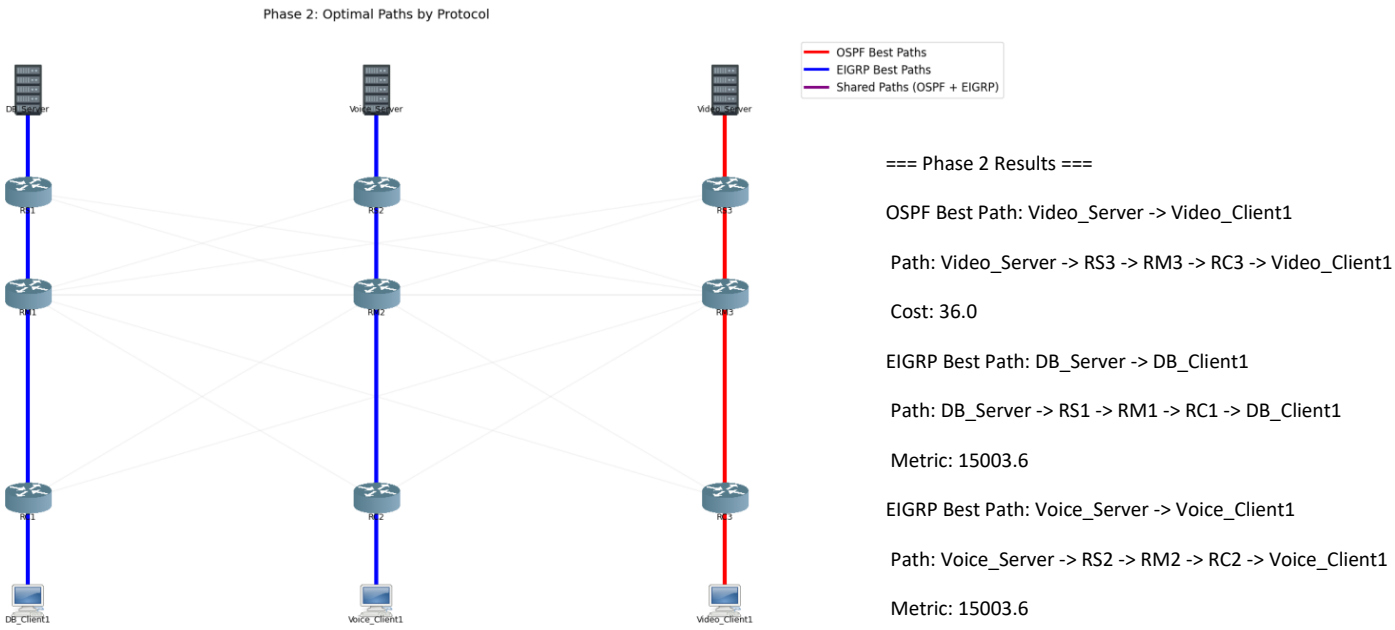
Ant Ant-DB_Server-DB_Client1-1755093345.4554975 finished. Success=True
Forward paths: ['DB_Server', 'RS1', 'RM3', 'RS3', 'RM2', 'RC3', 'RM1', 'RC1', 'RS2', 'DB_Client1']
Return paths: ['RS1', 'DB_Server', 'DB_Client1', 'RS2', 'RC1', 'RM1', 'RC3', 'RM2', 'RS3', 'RM3']
Ant Ant-DB_Server-DB_Client1-1755093345.4554975 counted as completed and valid.
```

الشكل (7-5)- جزء من محاكاة المرحلة الأولى

عند تشغيل المحاكاة نتج عن المرحلة الأولى (مرحلة الاستكشاف) المسارات الموضحة بالشكل (7-6) والتي تحدد المسارات المناسبة لكل بروتوكول حسب نوع البيانات المنقولة بالشبكة، والتي سيتم تطبيق بروتوكولات التوجيه المناسبة عليها في المرحلة الثانية ولينتج عنها لاحقاً المسار الأمثل لنقل كل نوع بيانات وذلك كما بالشكل (7-7).



الشكل (7-6)- اختيار المسارات في المرحلة الأولى



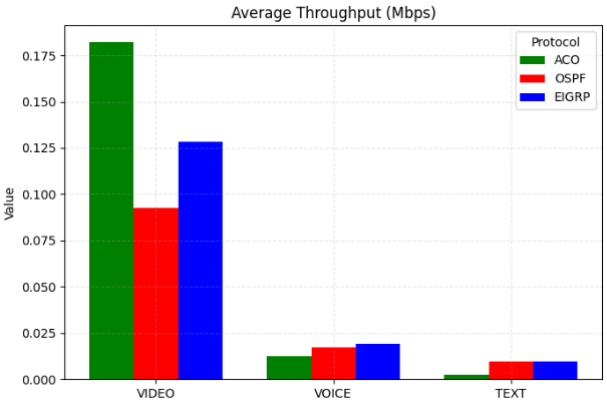
الشكل (7-7)- اختيار المسارات في المرحلة الثانية

- تم تطبيق البروتوكولات المناسبة على المسارات المكتشفة بواسطة الخوارزمية وقياس المحددات (زمن التأخير – الإنتاجية – التذبذب – زمن الاستجابة)
- ثم تم تطبيق البروتوكولين OSPF و EIGRP كل على حدة على نفس الشبكة وبدون تدخل الخوارزمية ومن ثم قياس (زمن التأخير – الإنتاجية – التذبذب – زمن الاستجابة)
- ثم تمت مقارنة نتائج تطبيق الخوارزمية مع نتائج تطبيق البروتوكولات بدون الخوارزمية فكانت النتائج كما هو موضح بالجدول (7–2):

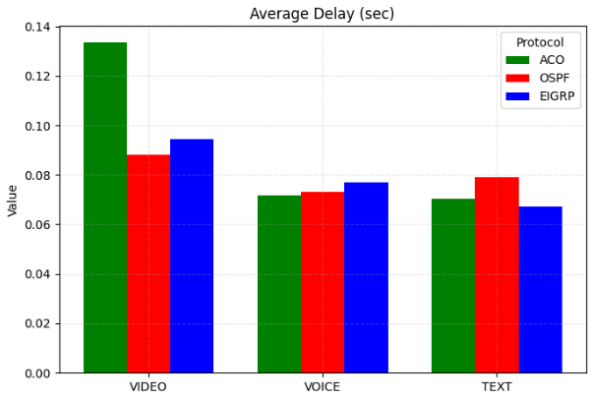
الجدول (7–2)– نتائج مقارنة الخوارزمية المقترحة مع نتائج بروتوكولات التوجيه

Metric (Data Type)	Modified ACO	OSPF	EIGRP
Delay (VIDEO)	0.13	0.09	0.09
Delay (VOICE)	0.07	0.07	0.08
Delay (TEXT)	0.07	0.08	0.07
Throughput (VIDEO)	0.18	0.09	0.13
Throughput (VOICE)	0.01	0.02	0.02
Throughput (TEXT)	0.00	0.01	0.01
Jitter (VIDEO)	0.02	0.01	0.01
Jitter (VOICE)	0.01	0.02	0.01
Jitter (TEXT)	0.01	0.01	0.01
RTT (VIDEO)	0.27	0.18	0.19
RTT (VOICE)	0.14	0.15	0.15
RTT (TEXT)	0.14	0.16	0.13

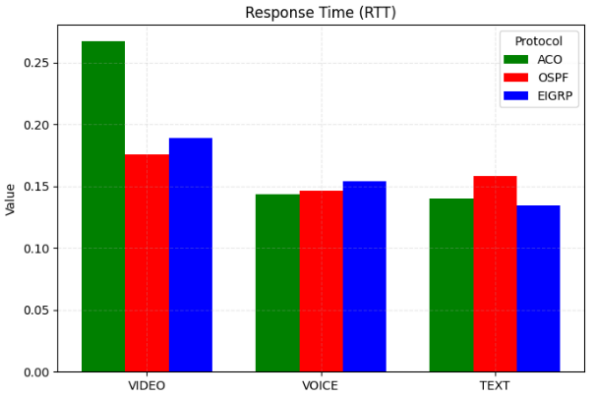
ويمكن إنشاء المخططات البيانية المعبّرة عن الجدول السابق كما يلي:



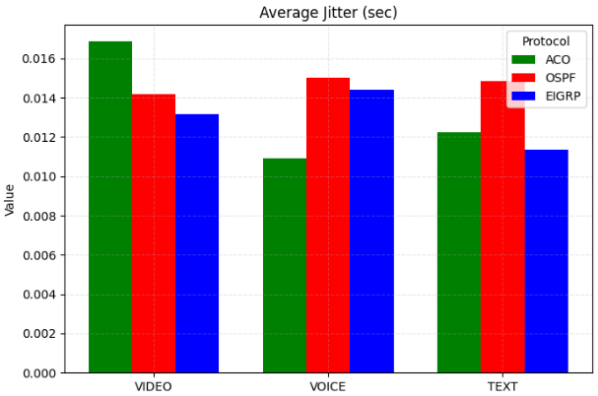
الشكل (7–8)– قياس الإنتاجية للخوارزمية المقترحة



الشكل (7–9)– قياس التأخير للخوارزمية المقترحة



الشكل (7-10)- قياس زمن الاستجابة للخوارزمية المقترحة



الشكل (7-11)- قياس التذبذب للخوارزمية المقترحة

في نهاية المرحلتين الأولى والثانية نتج لدينا ما يلي:

- عدد النمل المُولّد في المرحلة الأولى: 222
- النمل المُكتمل: 72
- النمل الفاشل: 150
- معدل النجاح: 32.43%
- قرارات بروتوكول التوجيه:
  - OSPF: 22
  - EIGRP: 50

7.7. الاستنتاجات

- الملاحظ من النتائج الأداء المتشابه في بعض المحددات عند تطبيق الخوارزمية مقارنة مع تطبيق البروتوكولات مباشرة على نفس الشبكة بدون تدخل الخوارزمية مع إنتاجية أعلى للفيديو من البروتوكولات المطبقة.
  - نقاط الضعف عند تطبيق الخوارزمية:
    - معامل الإنتاجية منخفض عند نقل الصوت و النصوص
    - معدل فشل النمل مرتفع

- الاقتراح:

  - استكشاف المسارات بعمق والتعامل مع الحلقات من أجل معالجة فشل النمل بالوصول للهدف
  - تقييم جودة الوصلات لزيادة الدقة

لذلك كان لابد من تلافي نقاط الضعف الخوارزمية بتطويرها من أجل رفع إنتاجيتها ومعدل نجاح النمل في اكتشاف المسارات ورفع كفاءتها.

8.7. التطوير الأول للخوارزمية المقترحة

1.8.7. إضافة الآلية DFS (Depth-First-Search):

تم دمج DFS بإعادة صياغتها ضمن خوارزمية ACO المطورة في هذه الدراسة لتحقيق استكشاف شامل للمسارات، مع الحفاظ على خصائص ACO وذلك كما يلي:

- **استخدام مكدس DFS داخل النملة:**

  - كل نملة تمتلك مكدس DFS لتخزين العقد التي تزورها وجيرانها.
  - عند الوصول إلى عقدة، يتم دفعها إلى المكدس مع قائمة جيرانها غير المزارين.
  - إذا انتهى الجيران المتاحين، يتم التراجع إلى العقدة السابقة (Backtrack) تماماً كما في DFS

- **إزالة الانحياز الاحتمالي**

  - عند اختيار الجار التالي، يتم ترتيب الجيران عشوائياً ولكن بدون أي تفضيل في الاحتمالات هذا يضمن أن كل المسارات لها فرصة متساوية في الاكتشاف.

- **تسجيل المسارات الكاملة**

  - كل نملة تحتفظ بالمسار الكامل لتخزين المسارات المكتشفة.

- يتم التحقق من طول المسار وعدد مرات التكرار قبل حفظه.

- التراجع الذكي عند الوصول إلى طريق مسدود

- إذا لم تجد النملة جيراناً غير مزارين، يتم التراجع إلى العقدة السابقة.
- يتم حذف العقدة من المسار الحالي وتحديث العقدة الحالية، كما في DFS التقليدية.

- منع الحلقات

- يتم تتبع عدد مرات زيارة كل عقدة لمنع الدوران في حلقة.
- إذا زارت النملة عقدة أكثر من مرتين، يتم التعامل معها كحلقة ويتم التراجع.

- التفاعل مع ACO المطورة

- بعد انتهاء مرحلة الاستكشاف، يتم حساب الفيرمون بناءً على نوع البيانات (data_type) والمسار المكتشف.
- يتم اتخاذ قرار البروتوكول (OSPF) أو (EIGRP) في مرحلة القرار بناءً على الفيرمون المتراكم.

1.1.8.7. ميزات الدمج:

إن دمج DFS داخل ACO المطوّرة يحقق ما يلي:

- اكتشاف شامل للمسارات بدون فقدان أي احتمالية.
- حفظ المسارات المتعددة لتقييمها لاحقاً.
- توازن بين الاستكشاف العشوائي (ACO) والاستكشاف المنظم (DFS)
- منع الانحياز وضمان عدالة في اختيار المسارات.

- تحسين واقعي في محاكاة الشبكات المعقدة.

2.1.8.7. الفرق بين DFS التقليدية والمطبقة داخل ACO المطوّرة:

يوضح الجدول التالي الفروق الأساسية كما يلي:

الجدول (7−3)− مقارنة بين خوارزمية DFS التقليدية وبين المطبقة داخل ACO المطورة

العنصر	DFS التقليدية	التطبيق داخل ACO المطورة
بنية المسار	تستخدم مكدس بسيط لتتبع المسار	تستخدم مكدس يحتوي على (العقد، الجيران)
التعامل مع المسارات	تستكشف مسار واحد بكل عمق ممكن	تستكشف عدة مسارات، وتحفظها بكل نملة
الحلقات	يمكن الوقوع في حلقات	يتم تتبعها ومنع زيارة العقد أكثر من مرتين
المسار النهائي	يُعيد المسار فور الوصول للهدف	يمر بمراحل (العودة و القرار) ومن ثم حساب الفيرمون
هدف البحث	الوصول إلى هدف معين	استكشاف كل المسارات الممكنة بين المصدر والهدف
الانتقال بين الجيران	ترتيب منطقي	ترتيب الجيران عشوائياً لضمان عدالة الاستكشاف

2.8.7. إضافة تقييم جودة الوصلات:

1.2.8.7. آلية تحديث الفيرمونات وتكيّفها مع الجودة

عند كل دورة تحديث في الخوارزمية المقترحة القائمة على عمل النمل الاصطناعي، يتم تنفيذ سلسلة مترابطة من الخطوات تضمن التكيّف الديناميكي للمسارات وفقاً للجودة الفعلية للشبكة. لقد تم إعادة تصميم آلية تحديث الفيرمون لتصبح واعية لجودة الخدمة (QoS-aware pheromone update) خلافاً للنموذج التقليدي في خوارزمية ACO حيث يُحدّث الفيرمون بشكل ثابت أو عشوائي.

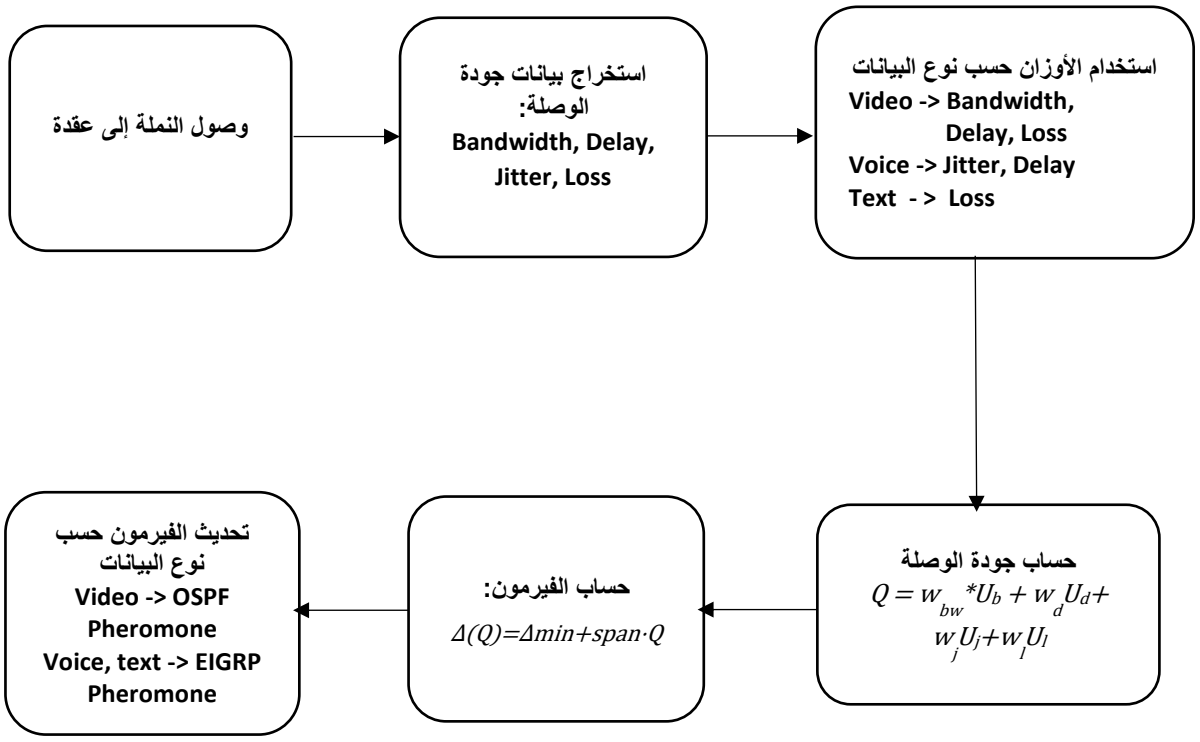
إن آلية التحديث الجديدة تعتمد على تقييم ديناميكي لجودة الوصلة بين العقد باستخدام مجموعة من المحددات تشمل: عرض الحزمة، التأخير، الفقد، والتذبذب.

2.2.8.7. حساب جودة الخدمة (Q) :

في البداية يتم تطبيع هذه المعايير ضمن المجال [0.0 – 1.0] ثم دمجها باستخدام أوزان خاصة لكل نوع بيانات (فيديو، صوت، نص) بما يراعي احتياجات كل خدمة، ومن تم تطوير آلية الفيرمون بحيث:

- تعكس جودة الوصلات بشكل ديناميكي.
- تميز بدقة بين احتياجات أنواع البيانات.
- تسمح ببناء مسارات متكيفة مع جودة الوصلات الواقعية.
- تُتخذ قرارات البروتوكول في مرحلة القرار بناءً على تراكم منطقي.

وذلك كما بالشكل (7−13) والذي يوضح حساب الجودة ضمن الخوارزمية المقترحة:



الشكل (7−13)− مخطط حساب الجودة في الخوارزمية المقترحة

- في النماذج التقليدية لحساب مؤشر الجودة الكلي(QoS) ، يتم استخدام صيغة منفعة متعددة المعايير (MAUT) ذات طبيعة أُسّية، بحيث تُدمَج القيم المطَبَّعة لمحددات الشبكة في مؤشر واحد:

$$Q = w_b \cdot U^*(u_b) + w_d \cdot U^*(u_d) + w_j \cdot U^*(u_j) + w_l \cdot U^*(u_l) \quad [27][28]$$

$$U^*(u) = 1 - e^{-a\,u}, \quad a > 0,\ u \in [0,1]$$

- U_b, U_d, U_j, U_l : القيم المطَبَّعة ضمن المجال [0 , 1] لمحددات جودة الخدمة (عرض الحزمة، التأخير ، التذبذب، الفقد) حيث تم التطبيع حسب ITU .
- w_b, w_d, w_j, w_l : الأوزان النسبية لكل محدد (وفق توصيات ITU) [29]
- a_i : معاملات شكلية تتحكم في انحناء دالة المنفعة (Exponential Utility Function) .

- هذه الصيغة تعكس خاصية العائد المتناقص (Diminishing Returns) ، حيث أن زيادة المنفعة تصبح أبطأ مع اقتراب المحدد من قيمته المثالية (أي أن تحسين الجودة عند المستويات العالية يكون أصعب وأقل تأثيراً).

3.2.8.7. الصيغة المعدلة مع معاملات (k_i):

- لزيادة المرونة والتحكم الديناميكي، قمنا بإدخال معاملات إضافية ki لكل محدد فأصبحت المعادلة المقترحة:

$$Q = w_b\, U_b^*(k_b) + w_d\, U_d^*(k_d) + w_j\, U_j^*(k_j) + w_l\, U_l^*(k_l)$$

$$U_i^*(k_i) = 1 - e^{-k_i\, a_i\, U_i}$$

- k_i : ∋ [1.0] معاملات معايرة ديناميكية يمكن تعديلها أثناء التشغيل، بحيث تضبط حساسية كل محدد للتغيرات في الشبكة.

1.3.2.8.7. مبدأ العمل:

- تُحدّث القيم اللحظية للمحددات (مثل التأخير والتذبذب ...) للحصول على تقدير مستقر للأداء ثم تتم مقارنة هذه القيم مع القيم المرجعية الموصى بها من قبل الاتحاد الدولي

للاتصالات (ITU-T) بحيث تُحدد الفجوات بين الأداء المرصود والقيم المثالية. واستناداً إلى هذه الفجوات، يتم تعديل معاملات المعايرة k_i بشكل ديناميكي، فإذا كان التأخير مثلاً أعلى من المستوى المطلوب، يتم خفض k_d لتخفيض قيمة الجودة عند حساب Q أما إذا كان الفقد مرتفعاً، يتم خفض k_l بنفس المبدأ.

إن هذا التكييف يجعل دالة الجودة أكثر حساسية للعوامل التي تعاني من التدهور.

2.3.2.8.7. فوائد إضافة k_i :

- **معايرة مرنة :** في الصيغة الأساسية، يعتمد شكل المنحني فقط على *ai*، والذي غالباً ما يُحدد مسبقاً ولا يتغير. بإضافة *ki*، يصبح بالإمكان ضبط المنحني ديناميكياً أثناء المحاكاة أو التشغيل الفعلي.
- **تكيّف مع الظروف :** يمكن للنظام خفض *ki* لمحدد (مثل التأخير في تطبيقات الصوت) عندما تتدهور الظروف، مما ينقص قيمة الجودة.
- **تحكم ديناميكي :** تتيح هذه المعاملات آلية "تعلّم" أو "تكيّف"، حيث تُحدث قيم ki استناداً إلى القياسات الفعلية للشبكة (مثل المتوسط والانحراف المعياري للجودة).
- **دقة أعلى في الموازنة :** وجود ki يسمح للنظام بالوصول إلى نقطة توازن واقعية بين المحددات، بدلاً من الاعتماد فقط على أوزان ثابتة *wi*

المعادلة الأساسية (بدون *ki*) تعطي إطاراً ثابتاً وموثقاً في الأدبيات لدمج مقاييس QoS باستخدام المنفعة الأُسّية.

أما الصيغة المعدلة (مع *ki*) فهي امتداد عملي مبتكر يضيف إمكانية التكيّف الذاتي والمعايرة الديناميكية، مما يجعلها مناسبة أكثر لتطبيقات المحاكاة والخوارزميات التكيفية مثل توجيه النمل (ACO-based routing) .

4.2.8.7. تحديث الفيرمونات:

تؤدي التغييرات في معاملات ki إلى تعديل القيم المحسوبة لـ Q وبما أن تحديث الفيرمونات يعتمد على Q، فإن توزيع الفيرمونات على الوصلات يتغير أيضاً. النتيجة النهائية هي أن احتمالات اختيار الوصلات تتبدل بمرور الوقت، مما يقود إلى تكيّف ديناميكي تلقائي للمسارات بحيث ينجذب النمل (Packets/Agents) نحو الوصلات ذات الأداء الأفضل فعلياً.

5.2.8.7. قياس الفيرمون:

تعتمد خوارزميات التوجيه المستوحاة من سلوك مستعمرات النمل (Ant Colony Optimization – ACO) على مبدأ تراكم الفيرمون كآلية غير مباشرة للتواصل بين الوكلاء (النمل) واختيار المسارات المثلى. حيث في النماذج الكلاسيكية، يُحدَّث الفيرمون عبر إضافة ثابتة مرتبطة بطول المسار أو تكلفته، مع تضمين عامل تبخر لتجنب التشبع. إلا أنه يختلف في دراستنا هذه حيث يرتبط الفيرمون ببروتوكولات التوجيه بالإضافة لارتباطه بجودة الخدمة (QoS) ومعاييرها المتعددة مثل: عرض الحزمة، التأخير، التذبذب، ومعدل الفقد. كما أن النهج الكلاسيكي يُعطي معاملة متجانسة لجميع الوصلات، حيث يعتمد على معاملات ثابتة (مثل الجودة وعامل التبخر)، ولا يأخذ في الحسبان ديناميكية ظروف الشبكة أو تباين جودة الخدمة بين الوصلات.

اقترحنا في هذه الدراسة إدخال معادلات ديناميكية تجعل الزيادة في الفيرمون مرتبطة بشكل مباشر بمستوى الجودة المقاس فعلياً. وتتمثل الصياغة المقترحة بما يلي:

- $\Delta(Q)=\Delta\min+(\Delta\max-\Delta\min)\cdot Q$
- $\Delta\min = \alpha\cdot(1-q_set)$
- $\Delta\max = \beta\cdot(1+\sigma Q)$

حيث تشير الرموز إلى ما يلي:

Δ(Q): هو الزيادة الفعلية في الفيرمون عند هذه الخطوة، فإذا كان Q عالياً، تكون الزيادة أكبر، مما يعزز البروتوكول، وإذا كان Q منخفضاً، تكون الزيادة صغيرة، مما يحد من تراكم الفيرمون.

Δmin: هو الحد الأدنى الممكن للزيادة في الفيرمون، يضمن أنه حتى في أسوأ الحالات (جودة ضعيفة جداً) يبقى هناك زيادة صغيرة في الفيرمون بدل أن يتوقف تماماً، مما يمنع تلاشي بروتوكول أو مسار بشكل كامل، ليُبقي خيارات بديلة قائمة.

Δmax: هو الحد الأقصى الممكن للزيادة في الفيرمون، حيث يضع سقفاً على مقدار التعزيز حتى لا يحدث تشبع مفرط (over-saturation) لبروتوكول واحد، مما يحافظ على التوازن ويمنع انحيازاً مطلقاً لبروتوكول واحد فقط.

(Δmax−Δmin)·Q: هذا الجزء يجعل العلاقة بين الجودة Q والزيادة في الفيرمون خطية مضبوطة بين حدّين:

- فإذا كانت Q=0 يكون عندها Δ(Q)=Δmin
- وإذا كانت Q=1 يكون عندها Δ(Q)=Δmax
- أما بقية القيم، يكون Δ(Q) بين الحدين بشكل نسبي.

- q_set : هو القيمة المرجعية (النقطة المثلى التي تم استنتاجها من المخطط في الشكل (7−15))
  - إذا كانت q_set عالية (مثلاً 0.8): (1- q_set) صغيرة فإن Δmin تكون صغيرة لأننا أصلاً قريبون من جودة ممتازة ولسنا بحاجة إلى تعزيز دائم.
  - أما إذا كانت q_set منخفضة (مثلاً 0.3): (1- q_set) كبيرة فإن Δmin تكون أكبر فنرفع الحد الأدنى من الفيرمون حتى نشجع النمل على تقوية الوصلات وتعويض الجودة الضعيفة.
- يحدد α شدة التأثير.
- σQ الانحراف المعياري للجودة يخبرنا إذا كانت الجودة متذبذبة:

- فإذا $\sigma Q \approx 0$ فإن الجودة ثابتة $\Delta max \approx \beta \cdot 1$ ويكون مجال استجابة صغير.
- أما إذا كان σQ كبيراً (الجودة متذبذبة): فإن Δmax يكون أكبر فنعطي النظام مساحة أوسع للاستجابة بقفزات عند الحاجة.

- التذبذب الكبير يعني أننا بحاجة إلى مرونة أكبر (span أوسع).
- وقيمة β تحدد كم نسمح لهذه المرونة أن تتوسع.

يبين الشكل (7−14) مثالاً عن تغير قيمة الفيرمون عند تغير قيمة محددين من المحددات ضمن الدراسة وهما الضياع و التذبذب، حيث المنحنى (a) يوضح أن ارتفاع التذبذب يؤدي إلى تقليل معدل تعزيز الفيرمون (Δ) وهذا منطقي، لأن التذبذب العالي يقلل من موثوقية الخدمة، والخوارزمية تعكس ذلك مباشرة بخفض تعزيز الروابط في تلك الظروف.

والمنحني هنا مقعّر حيث في البداية وعندما (Jitter منخفض) يكون $\Delta(Q)$ عالياً ويشجع على استخدام المسار، ولكن مع ارتفاع Jitter ينخفض $\Delta(Q)$ بسرعة.

المنحنى (b) يظهر أن الخسائر تؤثر بقوة على تعزيز الفيرمون، حيث أنه حتى فقدان بسيط نسبياً يمكن أن يقلل $\Delta(Q)$ بشكل ملموس، وهذا يتماشى مع توصيات ITU بأن الفقد من أكثر المؤشرات حساسية في جودة الخدمة.

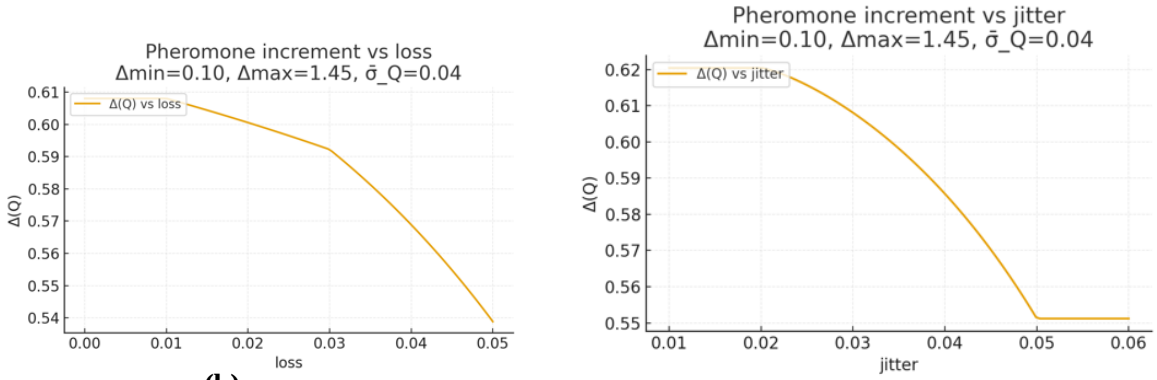
ومقارنةً بالتذبذب، فإن تأثير الفقد يكون أكثر حدة وذلك لأن خسارة الحزم تعني تدهوراً مباشراً في الكفاءة، فتتجاوب الخوارزمية بخفض الفيرمون سريعاً.

من المنحنيات في الشكل (7−14) نجد أن :

Jitter: تأثيره تدريجي، $\Delta(Q)$ تهبط تدريجياً مع زيادة التذبذب.

Loss: تأثيره أكثر حدّة، $\Delta(Q)$ يهبط أسرع بمجرد ارتفاع الخسارة.

هذا يجعل الخوارزمية أكثر صرامة مع الفقد، وأكثر تسامحاً نسبياً مع التذبذب.



(b) **(a)**

الشكل (7−14)− علاقة الفيرمون مع محددات مثل الضياع والتذبذب ضمن الشبكة

كما أن الشكل (7−15) يظهر منحنيات (عرض الحزمة، التأخير، التذبذب، الضياع) كل واحد منهم يبين قيمة الجودة (Q) حين نغير k الخاص به فقط ونثبت الثلاثة الآخرين، بالإضافة للمنحني البنفسجي المتقطع وهو المتوسط لمنحنيات المحددات الأربعة لالتقاط الجودة الكلية مقابل k، حيث تدل المنحنيات على ما يلي:

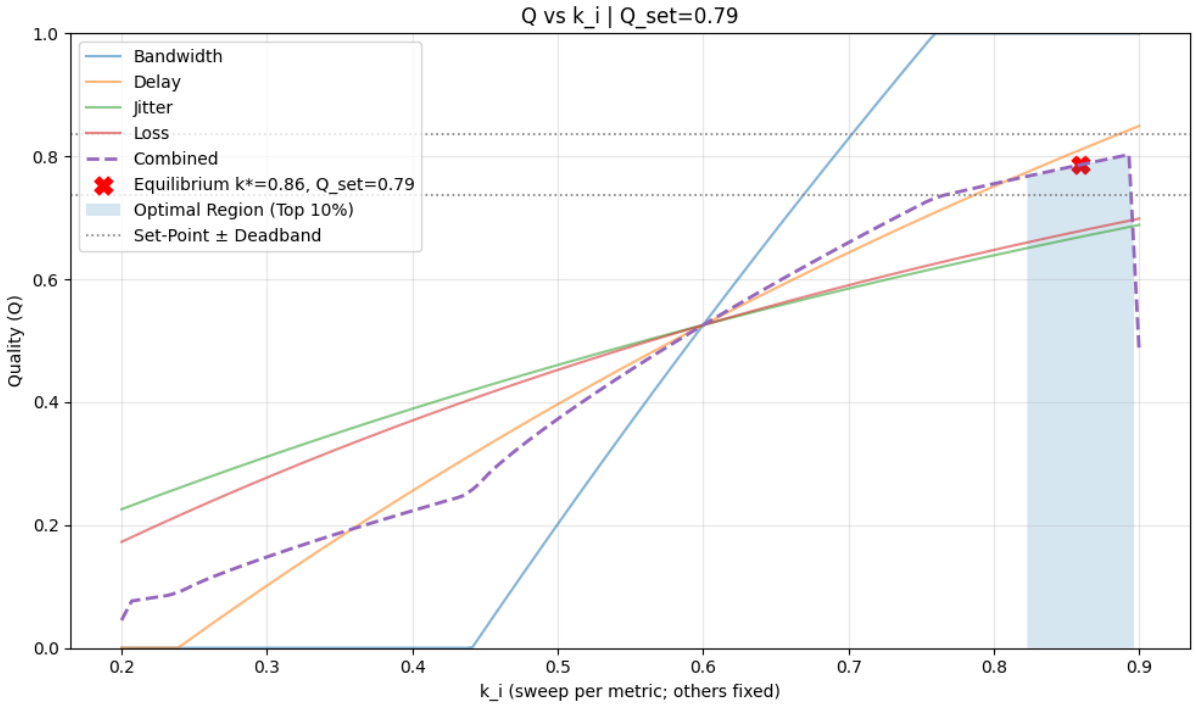
عرض الحزمة (Bandwidth): نلاحظ وجود صعود حاد بعد منتصف المجال؛ وذلك يدل أن زيادة حساسية عرض الحزمة k_b تعطي عوائد متزايدة في هذا النطاق وهذا يفسّر ميل المنحنى المجمّع للارتفاع مع اقتراب k من ~ 0.85 .

التأخير (Delay) و الضياع (Loss): نلاحظ نمو شبه خطي معتدل؛ تأثيرهما إيجابي لكن بعائد حدي أقل من عائد عرض الحزمة حسب المخطط.

التذبذب (Jitter): نمو متباطئ تدريجياً (محدّب/مقعّر بحسب α_j لذا مساهمته في رفع Q مع k_j أقل حدة من عرض الحزمة.

الهبوط قرب k≈0.90 في المنحني المتوسط: إشارة تفيد بأن الاندفاع إلى أقصى k قد لا يكون مثالياً دائماً (تأثير معايرة/تشبّع/تناسق بين المحددات)، ومن هنا جاءت فائدة أخذ مركز الكتلة داخل أعلى 10% بدل اختيار آخر نقطة.

النقطة الحمراء : تدل على نقطة التوازن Q_set = 0.79 والمنطقة الزرقاء هي منطقة العمل المثلى



الشكل (7-15)- علاقة k_i بالجودة وتحديد منطقة العمل

6.2.8.7. الأوزان والتطبيع (لكل خدمة، حسب ITU): [29]

يوضح الجدول (7-4) قيم الأوزان المستخدمة للخدمات ضمن هذه الدراسة وذلك حسب (IUT: International Telecommunication Union) للخدمات المختلفة.

الجدول (7-4)- أوزان الخدمات حسب ITU-T G.1010 [28]

الخدمة	w_b (BW)	w_d (Delay)	w_j (Jitter)	w_l (Loss)
فيديو	0.6	0.2	0.1	0.1
صوت	0.1	0.3	0.4	0.2
قواعد بيانات	0.1	0.2	0.0	0.7

- **Delay** (ms):

$$U_d = \begin{cases} 1 & d \leq 150 \\ 0 & d \geq 400 \\ 1 - \left(\frac{d-150}{250}\right)^{1.5} & 150 < d < 400 \end{cases}$$

- **Jitter** (ms):

$$U_j = \begin{cases} 1 & j \leq 20 \\ 0 & j \geq 50 \\ 1 - \left(\frac{j-20}{30}\right)^{1.3} & 20 < j < 50 \end{cases}$$

- **Loss** (%):

$$U_l = \begin{cases} 1 & l \leq 1 \\ 0 & l \geq 5 \\ \frac{l-1}{2} \cdot 0.3 & 1 < l \leq 3 \\ 0.3 + \left(\frac{l-3}{2}\right) \cdot 0.7 & 3 < l < 5 \end{cases}$$

- **Bandwidth**:

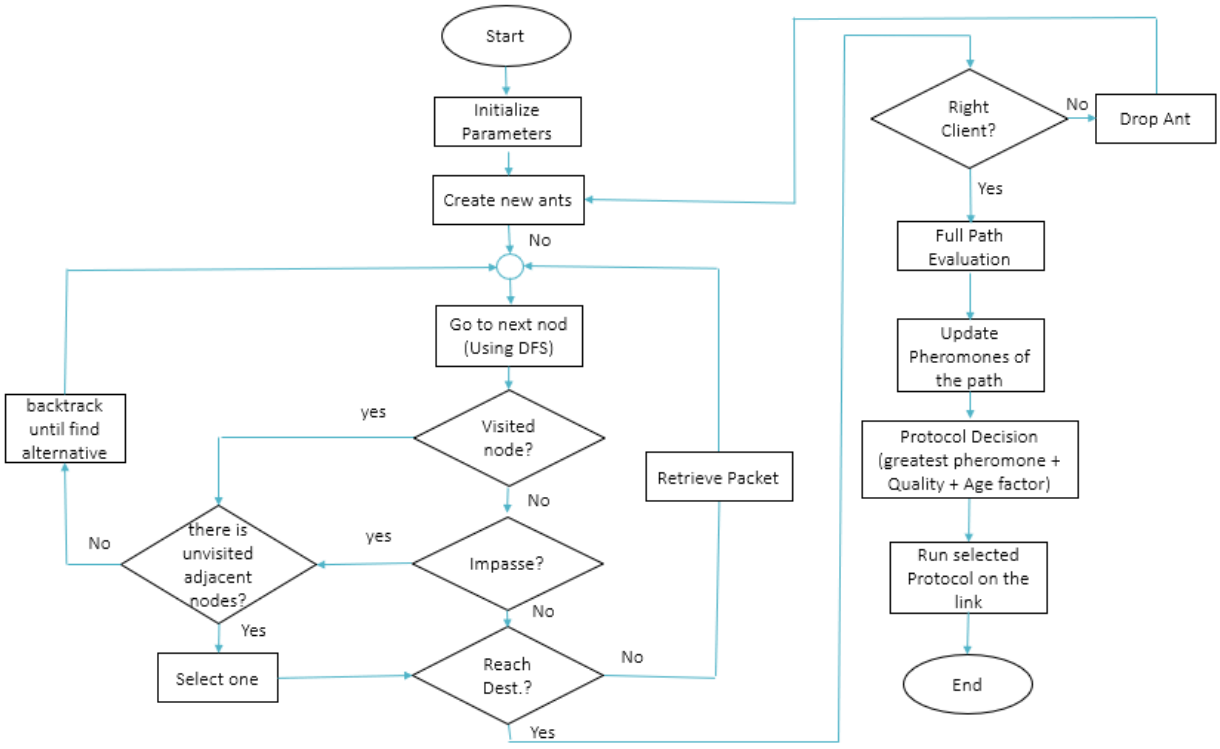
$$U_b = \min\left(1, \frac{bw}{\text{bw_cap}}\right)$$

where:

- $\text{bw_cap} = 4000$ kbps (Video),
- $\text{bw_cap} = 100$ kbps (Voice),
- $\text{bw_cap} = 50$ kbps (Text).

3.8.7. الخوارزمية المقترحة مع التطوير الأول:

يظهر المخطط في الشكل (7–16) الخوارزمية المقترحة بعد دمج الخوارزمية DFS ضمن مرحلة الاستكشاف وإضافة آلية التراجع عند وجود فشل في عقدة أو وصلة ما، بالإضافة لإدخال آلية تقييم جودة الوصلات في عملية اتخاذ القرار كما يلي:



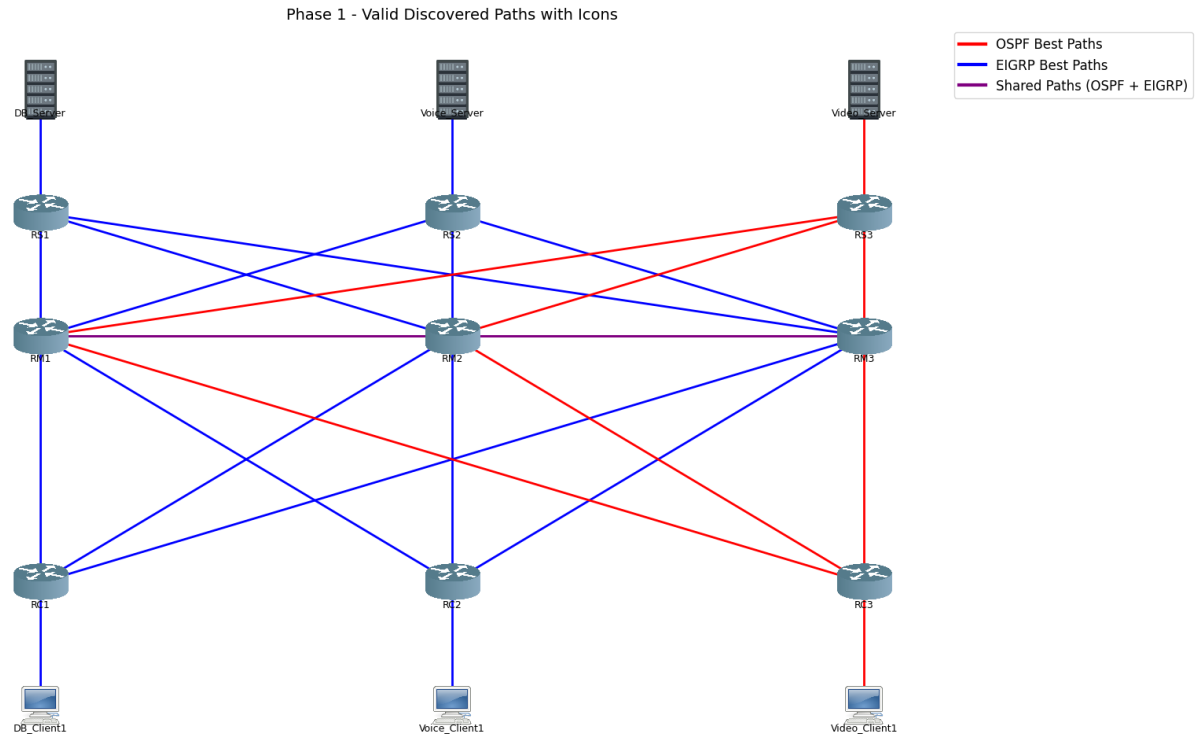
الشكل (7-16)- مخطط الخوارزمية المقترحة بعد التطوير الأول

4.8.7. نتائج المحاكاة

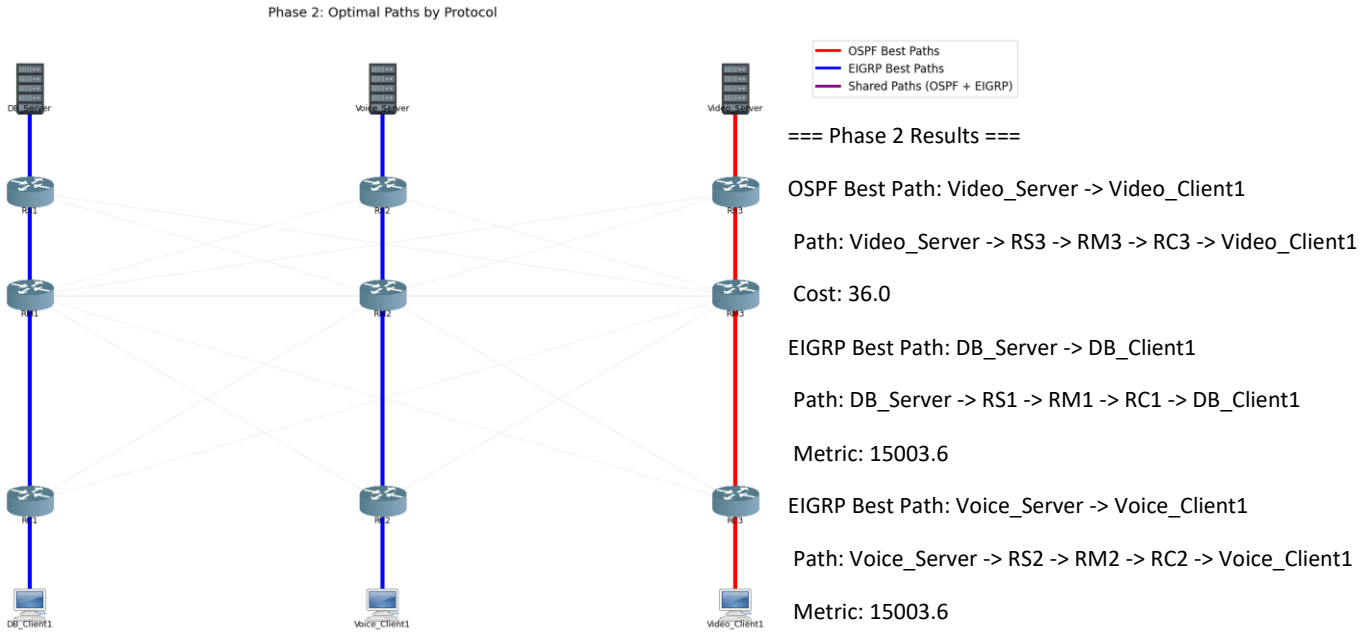
```
Ant Ant-Voice_Server-Voice_Client1-1756717137.1820347 counted as completed and valid.
Generated Ant from Video_Server → Video_Client1 (VIDEO)
OSPF pheromone += 0.67 → 1.67  [dmin=0.20, dmax=1.60, meanQ=0.34, stdQ=0.00]
OSPF pheromone += 0.96 → 2.62  [dmin=0.21, dmax=1.65, meanQ=0.32, stdQ=0.02]
OSPF pheromone += 0.96 → 3.59  [dmin=0.21, dmax=1.64, meanQ=0.31, stdQ=0.02]
OSPF pheromone += 0.94 → 4.53  [dmin=0.21, dmax=1.65, meanQ=0.31, stdQ=0.02]
OSPF pheromone += 0.96 → 5.49  [dmin=0.21, dmax=1.64, meanQ=0.30, stdQ=0.02]
Backtracked from Video_Server to RS3
OSPF pheromone += 0.94 → 6.43  [dmin=0.21, dmax=1.65, meanQ=0.30, stdQ=0.02]
OSPF pheromone += 0.93 → 7.36  [dmin=0.21, dmax=1.66, meanQ=0.29, stdQ=0.02]
OSPF pheromone += 0.92 → 8.28  [dmin=0.22, dmax=1.67, meanQ=0.29, stdQ=0.03]
OSPF pheromone += 0.91 → 9.19  [dmin=0.22, dmax=1.67, meanQ=0.28, stdQ=0.03]
OSPF pheromone += 0.91 → 10.11  [dmin=0.22, dmax=1.68, meanQ=0.28, stdQ=0.03]
OSPF pheromone += 0.91 → 11.02  [dmin=0.22, dmax=1.68, meanQ=0.27, stdQ=0.03]
OSPF pheromone += 0.91 → 11.93  [dmin=0.22, dmax=1.68, meanQ=0.27, stdQ=0.04]
OSPF pheromone += 0.92 → 12.85  [dmin=0.22, dmax=1.68, meanQ=0.26, stdQ=0.04]
OSPF pheromone += 0.92 → 13.77  [dmin=0.22, dmax=1.68, meanQ=0.26, stdQ=0.04]
OSPF pheromone += 0.92 → 14.68  [dmin=0.23, dmax=1.68, meanQ=0.26, stdQ=0.04]
OSPF pheromone += 0.92 → 15.60  [dmin=0.23, dmax=1.68, meanQ=0.26, stdQ=0.03]
OSPF pheromone += 0.92 → 16.52  [dmin=0.23, dmax=1.68, meanQ=0.26, stdQ=0.03]
OSPF pheromone += 0.92 → 17.44  [dmin=0.23, dmax=1.68, meanQ=0.25, stdQ=0.03]
OSPF pheromone += 0.98 → 18.42  [dmin=0.23, dmax=1.68, meanQ=0.26, stdQ=0.03]
OSPF pheromone += 0.96 → 19.38  [dmin=0.23, dmax=1.68, meanQ=0.26, stdQ=0.03]
OSPF pheromone += 1.00 → 20.38  [dmin=0.23, dmax=1.68, meanQ=0.26, stdQ=0.03]
```

الشكل (7-17)- جزء من محاكاة المرحلة الأولى

في نهاية المرحلة الأولى من المحاكاة نتج الشكل (7-18) والذي يبين المسارات المكتشفة والمحددة لكلا البروتوكولينOSPF و EIGRP، أما الشكل (7-19) فيبين نتيجة محاكاة المرحلة الثانية:



الشكل (7-18)- اختيار المسارات في المرحلة الأولى



الشكل (7-19)- اختيار المسارات في المرحلة الثانية

أما نتيجة المرحلتين الأولى والثانية فكانت كما يلي:

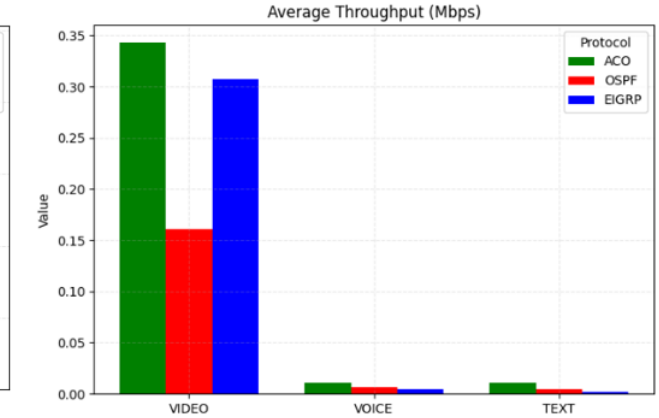
- النمل المُولّد في المرحلة الأولى: 222
- النمل المُكتمل: 222
- النمل الفاشل: 0
- معدل النجاح: 100%
- قرارات بروتوكول التوجيه:
  - OSPF: 74
  - EIGRP: 148

في البداية تم تطبيق الخوارزمية وقياس المحددات التالية: (زمن التأخير – الإنتاجية – التذبذب – زمن الاستجابة) ثم تم تطبيق البروتوكولين OSPF و EIGRP كل على حدة على نفس الشبكة وبدون تدخل الخوارزمية ومن ثم قياس (زمن التأخير – الإنتاجية – التذبذب – زمن الاستجابة).

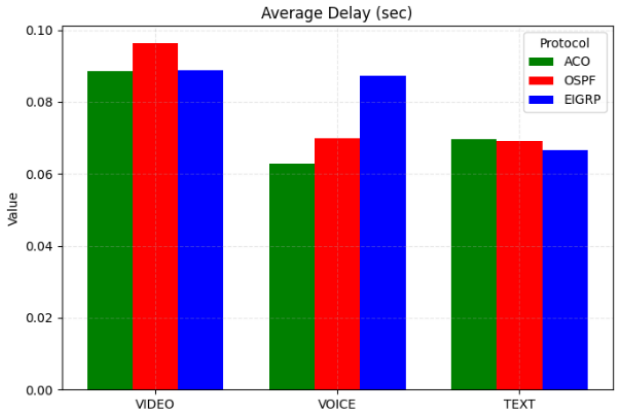
ثم تمت مقارنة نتائج تطبيق الخوارزمية مع نتائج تطبيق البروتوكولات بدون الخوارزمية فكانت النتائج كالتالي:

الجدول (7–5)– مقارنة الخوارزمية المقترحة المطورة مع نتائج بروتوكولات التوجيه

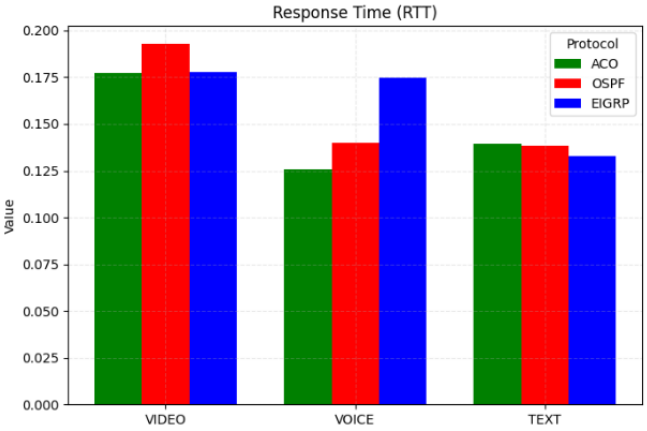
Metric (Data Type)	ACO	OSPF	EIGRP
Delay (VIDEO)	0.09	0.10	0.09
Delay (VOICE)	0.06	0.07	0.09
Delay (TEXT)	0.07	0.07	0.07
Throughput (VIDEO)	0.34	0.16	0.31
Throughput (VOICE)	0.01	0.01	0.00
Throughput (TEXT)	0.01	0.00	0.00
Jitter (VIDEO)	0.01	0.01	0.02
Jitter (VOICE)	0.01	0.01	0.01
Jitter (TEXT)	0.01	0.01	0.01
RTT (VIDEO)	0.18	0.19	0.18
RTT (VOICE)	0.13	0.14	0.17
RTT (TEXT)	0.14	0.14	0.13



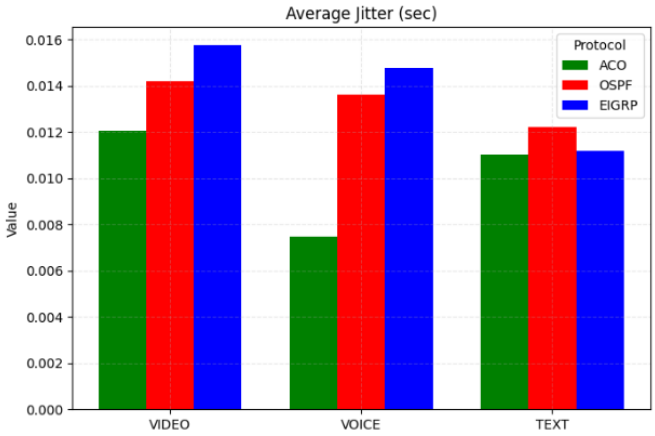
الشكل (7-20)- قياس الإنتاجية بعد التطوير الأول



الشكل (7-21)- قياس التأخير بعد التطوير الأول



الشكل (7-22)- قياس زمن الاستجابة بعد التطوير الأول



الشكل (7-23)- قياس التذبذب بعد التطوير الأول

5.8.7. الاستنتاجات

- من الجدول (7-5) والأشكال من (7-20) حتى (7-23) نجد تفوق الأداء عند تطبيق الخوارزمية بعد التطوير لعدد من محددات قياس البروتوكولين المطبقين على الشبكة بدون تدخل الخوارزمية، وتقاربه من الآخر فكان لها الأداء الأفضل بشكل عام.
- تحسن معامل الإنتاجية للشبكة بنسبة (89%) عما كان قبل التطوير وتفوقه على البروتوكولين أيضاً.
- ارتفاع ملحوظ بمعدل نجاح النمل بالوصول للهدف من 32.43% إلى 100%

6.8.7. المقارنة بين الخوارزمية ACO المطورة والكلاسيكية

التعديلات التي تم إجراؤها على الخوارزمية الكلاسيكية:

لقد تم إجراء العديد من التعديلات على خوارزمية ACO الكلاسيكية وتم تصنيفها حسب ما يلي:

الجدول (7-6)- مقارنة بين ACO الكلاسيكية و ACO المقترحة بعد التطوير الأول

العنصر	ACO الكلاسيكية	ACO المطوّرة
الهدف الأساسي		
الهدف	إيجاد أقصر مسار من المصدر للهدف	تحديد البروتوكول الأنسب (OSPF أو EIGRP) لكل مسار
معيار التقييم	الطول/الوزن أو الزمن	جودة الخدمة :(QoS) تأخير، عرض نطاق، فقد حزم ، انتاجية
التحكم في حركة النمل		
طريقة اختيار العقدة التالية	تعتمد على احتماليات مدروسة من الفيرمون	يتم إلغاء الانحياز الاحتمالي — توزيع متساوٍ أو عشوائي
التكرار	يمكن للنملة أن تزور العقد نفسها مراراً	لا يُسمح بزيارة العقدة أكثر من مرتين
تبخر الفيرمون	مفعّل ويحدث دورياً	معطّل تماماً – كل القيم محفوظة بدون فقد
دورة حياة النملة		
استكشاف (Exploration)	من المصدر إلى الهدف	نعم، بنفس الطريقة مع شروط صارمة للخدمة
عودة (Return)	غالباً غير موجود أو اختياري	موجودة، وتُستخدم لحساب الفيرمون بدقة
نشر القرار (Decision)	غير موجود	يتم إرسال النملة مرة ثالثة لتفعيل البروتوكول المناسب في كل راوتر
آلية الفيرمون		
عدد أنواع الفيرمون	فيرمون واحد فقط	فيرمونين: واحد لـ OSPF وآخر لـ EIGRP
موقع التحديث	عند كل قفزة	فقط بعد تحليل المسار الكامل، يتم تحديث المسار كاملاً
طريقة التقييم	طول المسار فقط	تحليل متوسط: delay, jitter, bandwidth, loss

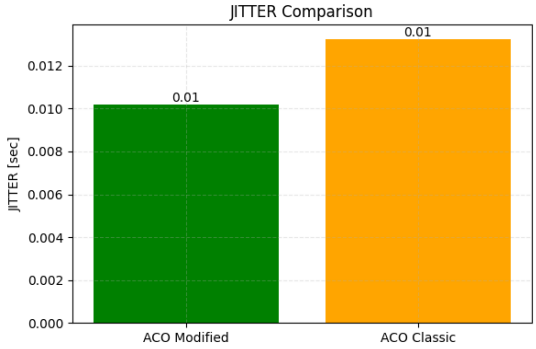
تأثير الفيرمون	يزيد احتمال اختيار العقد	يستخدم فقط لاتخاذ قرار نوع البروتوكول وليس للمسار
تشكل الحلقات		
اكتشاف الحلقة أثناء المسار	غير مطبق غالباً	عند تكرار عقدة أكثر من مرتين فإن النملة تعلن حلقة وتعود
رد الفعل عند اكتشاف الحلقة	إهمال أو فقد النملة	تستخدم آلية حذف آخر عقدة من المسار والعودة
استخدام DFS ضمن الاستكشاف		
نمط الحركة	يعتمد على الاحتمالية والفيرمون	يعتمد على مؤشر زمني + تتبع مسار كامل + backtracking
الانتقال للعقدة التالية	اختيار احتمالي موجه بالفيرمون	إزالة الانحياز، واختيار عشوائي غير موجه
معالجة الطريق المسدود	غير موجود	العودة للخلف وإعادة المحاولة
قيود الخدمة وتصفية المسارات		
مفهوم نوع البيانات	غير موجود	يطبق عند كل قفزة
تصفية المسارات	غير موجودة	يجب توافق السيرفر – العميل • لا يُسمح لسيرفر الفيديو بإرسال بيانات لعميل صوتي مثلاً. • النملة تُمنع من المرور بوصلات غير متوافقة بناءً على نوع الخدمة. عند العودة، يتم فلترة المسارات غير المتوافقة ولا تُسجّل نهائياً.

7.8.7. مقارنة نتائج الخوارزمية المطورة مع الكلاسيكية:

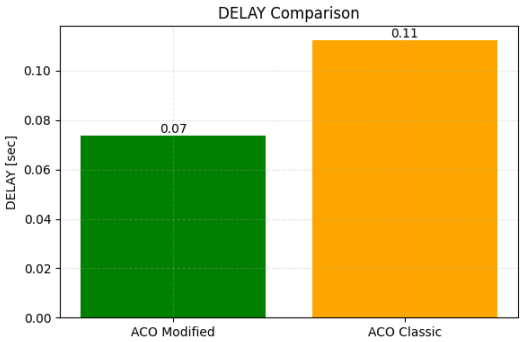
تم في الفقرة (5.8.7) مقارنة نتائج الخوارزمية المطورة مع البروتوكولين OSPF و EIGRP، وفيما يلي نتائج المقارنة بين نتائج الخوارزمية المطورة مع الكلاسيكية حسب المحددات التالية : (التأخير، التذبذب، زمن الاستجابة، الإنتاجية، وهي كالآتي:

الجدول (7-7)- مقارنة بين نتائج ACO الكلاسيكية وACO المقترحة بعد التطوير الأول

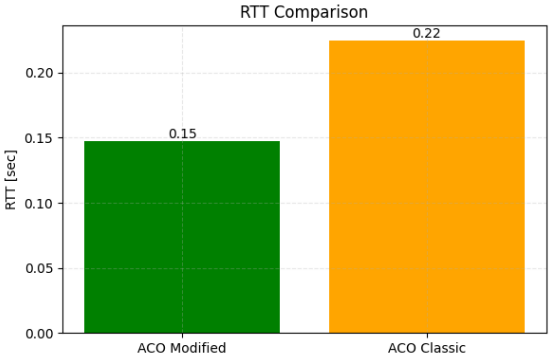
المحدد	المطورة ACO	الكلاسيكية ACO	التحسين %
التأخير	0.0737	0.1124	34.49 %
التذبذب	0.0102	0.0132	23.21 %
زمن الاستجابة	0.1473	0.2249	34.49 %
الإنتاجية	0.1216	0.1042	16.73 %



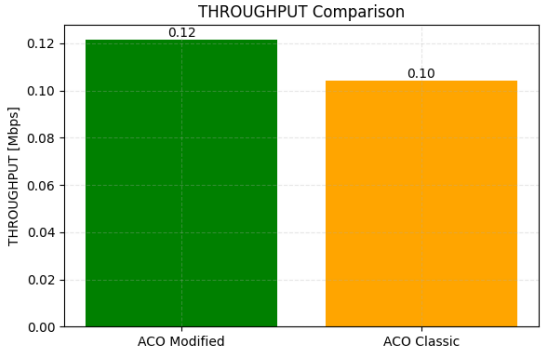
الشكل (7-24)- قياس التذبذب عند المقارنة مع الخوارزمية الكلاسيكية



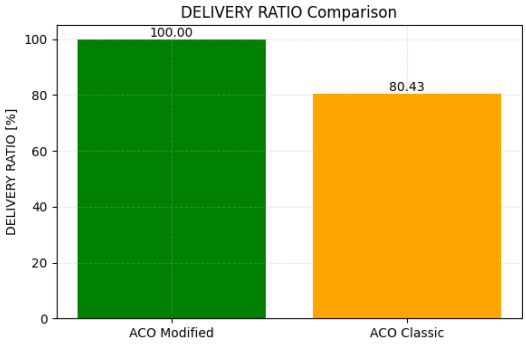
الشكل (7-25)- قياس التأخير عند المقارنة مع الخوارزمية الكلاسيكية



الشكل (7-26)- قياس زمن الاستجابة عند المقارنة مع الخوارزمية الكلاسيكية



الشكل (7-27)- قياس الإنتاجية عند المقارنة مع الخوارزمية الكلاسيكية



الشكل (7-28)- قياس نسبة التوصيل عند المقارنة مع الخوارزمية الكلاسيكية

8.8.7. تحليل النتائج:

من الجدول (7–7) والأشكال من (7–24) حتى (7–28) نجد ما يلي:

- تحسين التأخير (Delay) بنسبة 34.49% ، أي تقليل كبير في الزمن اللازم لوصول الحزم، مما ينعكس إيجابياً على جودة الخدمة.
- تحسين التذبذب (jitter) بنسبة 23.21% يؤكد أن الخوارزمية المطورة أكثر استقراراً في الإرسال.
- تحسين زمن الاستجابة (RTT) بنفس نسبة التأخير، مما يعزز الاعتمادية في التطبيقات بالزمن الحقيقي (مثل VoIP والفيديو).
- ازدادت الإنتاجية (Throughput) بنسبة 16.73%، أي أن الكفاءة في استخدام القناة ازدادت.
- ازدادت نسبة تسليم الحزم (Delivery Ratio) : فقد حققت الخوارزمية نسبة تسليم 100% وهي نسبة مرتفعة ومناسبة في الشبكات الحرجة حيث هذه النسبة تعد مؤشراً على جودة المسارات المختارة في الشبكة.

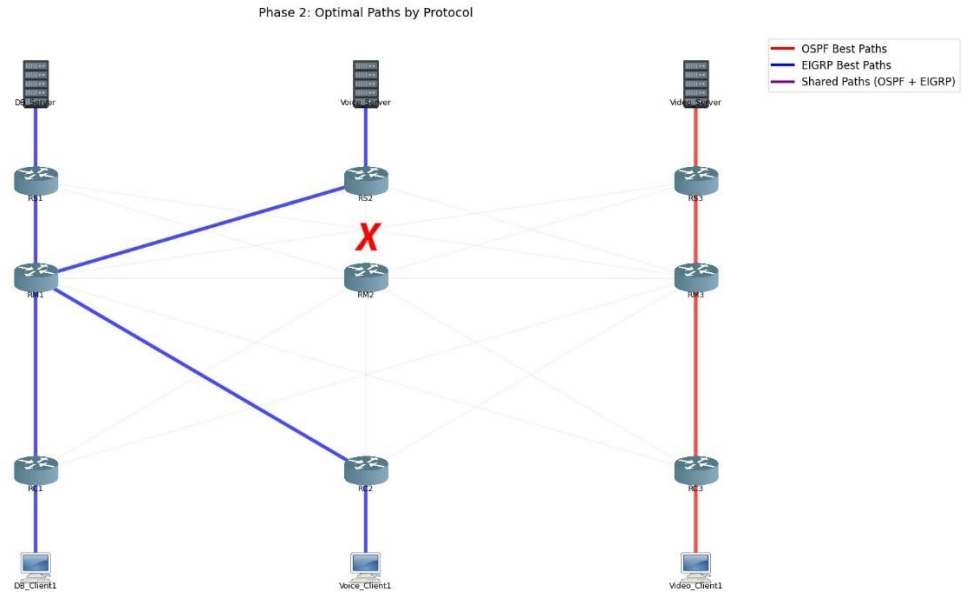
9.8.7. مقارنة مرجعية للمرجع (Shahid et *al.* 2024) [8] مع الخوارزمية المقترحة

- تم تشغيل البروتوكولات ضمن سيناريو يشمل: Load Sharing (تقاسم الحمل بين الوصلات) و Link Failover (التبديل التلقائي عند فشل الوصلة)
- تم إجراء مقارنة علمية بين نتائج الخوارزمية المقترحة المطورة في هذا البحث (ACO المطورة) وبين النتائج المنشورة في المقالة والتي تناولت دراسة أداء بروتوكولات التوجيه OSPF وEIGRP وBGP ضمن شبكة سلكية، مع التركيز على مقاييس الأداء مثل التأخير، التذبذب، نسبة التسليم، وزمن التقارب.

من أجل المقارنة، تم إجراء تعديل إضافي ضمن الخوارزمية المقترحة والمطورة في هذا البحث لتدعم خاصية Failover فأصبحت تتحقق في كل قفزة إن كانت الوصلة معطلة أم لا، هذا يعني أن الخوارزمية أصبحت "تأخذ قرارات واعية" بناءً على وضع الشبكة، وليست تسير فقط على الرسم البياني الثابت، وأيضاً

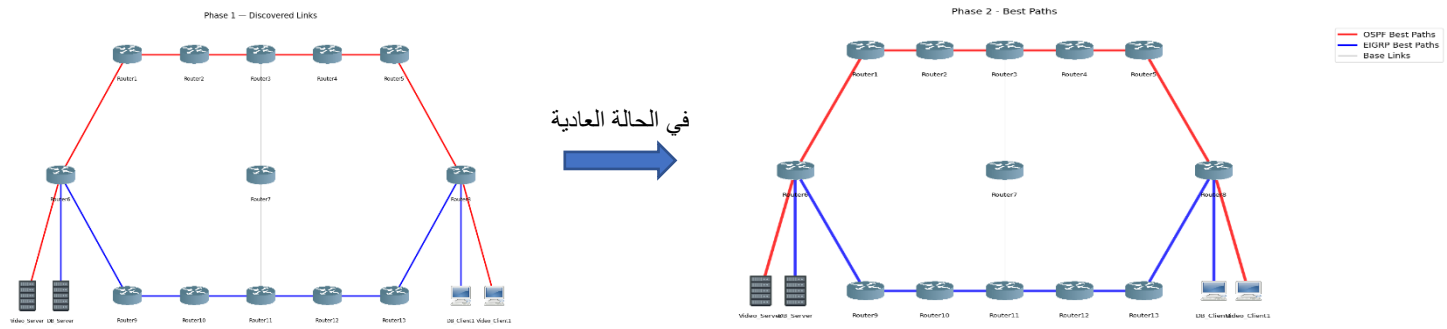
تم إضافة خاصية Load Sharing لتوزيع حركة النمل بناءً على الحالة الفعلية لكل وصلة أو عقدة، وليس فقط بناءً على التوجيه التقليدي. حيث يتم فيه:

- مراقبة الحمل الحالي على كل وصلة أو راوتر.
- توجيه الحزم إلى المسارات الأقل ازدحاماً أو الأقل تأخيراً.
- تعديل القرار في كل قفزة بناءً على الظروف اللحظية وليس فقط الفيرمون
- تعديل البروتوكولين OSPF و EIGRP ليدعما الآليتان : Failover و Load Sharing عند عملهما بشكل مستقل على الشبكة
- تم حذف الوصلة بين : RS2 و RM2 ، وبناءً عليه قامت الخوارزمية بتعديل المسارات السابقة وأصبحت كما يلي:



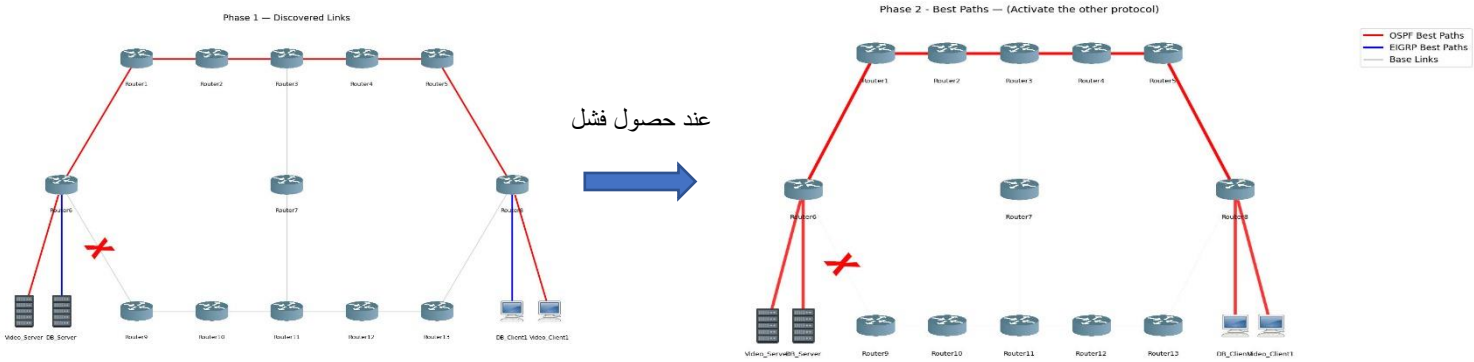
الشكل (7-29)- اختيار المسارات عند فشل وصلة

1.9.8.7. تطبيق الخوارزمية المقترحة على شبكة المرجع [8] من أجل المقارنة المرجعية:



(a) **(b)**

الشكل (7−30)− اختيار المسارات في الشبكة المرجعية بالحالة العادية



(a) **(b)**

الشكل (7−31)− اختيار المسارات في الشبكة المرجعية عند فشل وصلة

تم تطبيق الخوارزمية على الشبكة المدروسة بدون وجود فشل بالوصلات وذلك كما بالشكل (7−30) فكانت نتائج المرحلة الأولى باختيار المسارات المناسبة كما بالشكل (a) ومن ثم اتخذت المسارات المحددة بالشكل (b) في المرحلة الثانية، بعد ذلك تم إجراء المحاكاة بعد وجود فشل بالوصلة بين الموجهين (Router6, Router9) وذلك كما بالشكل (7−31)، فكانت نتيجة المرحلة الأولى كما يظهره الشكل (a) ومن ثم كانت نتيجة المرحلة الثانية كما بالشكل (b)، فنلاحظ أن الخوارزمية قد غيرت المسارات المحددة في الحالة الطبيعية بدون وجود فشل واتخذت المسارات المحددة بالشكل (7−31) عند حصول فشل بأحد الوصلات.

وقد تم قياس أداء الخوارزمية المقترحة مقارنة مع الخوارزمية الكلاسيكية وكلا البروتوكولين OSPF و EIGRP وسجلت النتائج بالجدول (7−8):

الجدول (7-8)- مقارنة بين أداء الخوارزمية المقترحة بعد التطوير وبين بروتوكولات التوجيه في الشبكة المرجعية

مقياس الأداء	ACO المطورة	ACO الكلاسيكية	EIGRP	OSPF	التحليل
Delay (sec)	0.270	0.312	0.317	0.366	الخوارزمية تحقق تأخيراً أقل
Jitter (sec)	0.029	0.034	0.035	0.032	أقل تذبذب، مناسبة للتطبيقات الحساسة
RTT (sec)	0.540	0.623	0.633	0.732	زمن الاستجابة أقل
Throughput (Mbps)	0.078	0.038	0.053	0.049	فعالية الخوارزمية المطورة أعلى
Packet Loss (%)	2.5	2.5	10	15	نسبة الفقد بالخوارزمية المطورة أقل بكثير

2.9.8.7. تحليل النتائج:

بتحليل النتائج التي حصلنا عليها نجد ما يلي:

- **تفوق ملحوظ في الأداء الزمني:**

الخوارزمية المقترحة تحقق تأخيراً أقل، واستجابة أسرع وتذبذباً منخفضاً جداً، ما يجعلها مثالية للتطبيقات الحساسة زمنياً (مثل الفيديو، الصوت، قواعد البيانات). وهذا يعود إلى استراتيجية ACO المطورة التي تكتشف المسارات مسبقاً بناءً على نوع الخدمة وتقييم المسار بشكل كامل.

- **تحكم ذكي في التوجيه:**

بخلاف الدراسة [8] التي تعتمد على قرارات التوجيه التقليدية، فإن الخوارزمية المطورة تدمج بين ACO وDFS، وتحدد المسارات بناءً على الفيرمون الخاص بكل بروتوكول، مع مراعاة خصائص الخدمة.

- **نسبة تسليم الحزم:**

حققت الخوارزمية نسبة تسليم للحزم 97.5 % وهي نسبة عالية ومناسبة لشبكات التطبيقات الحرجة حيث هذه النسبة تعد مؤشراً على جودة المسارات المختارة، بينما كانت النسبة للبروتوكولات في شبكة المقالة أقل بكثير.

– من النتائج نجد أن الخوارزمية المطورة المقترحة تتفوق بشكل واضح في تقليل زمن التأخير، التذبذب، ووقت الاستجابة، وهو ما يجعلها ملائمة جداً لشبكات الزمن الحقيقي وبيئات الضغط العالي.

تُثبت هذه المقارنة أن الخوارزمية المقترحة تحقق أداءً أفضل في عدة مؤشرات قياسية، مما يدعم جدوى تطبيقها في بيئات الشبكات الحقيقية، ويُعزز من القيمة التطبيقية للخوارزمية.

9.7. الخوارزمية المقترحة النهائية مع التطوير الثاني:

بعد التطوير الثاني والذي تم إضافة الخاصيتين failover و Load Sharing إلى الخوارزمية المقترحة، أصبحت الخوارزمية المقترحة النهائية بعد التطويرات تعمل وفق الخطوات التالية:

تهيئة المحددات الأولية للخوارزمية:

- تحديد عدد النمل في المستعمرة.
- تهيئة مستويات الفيرمون لكل بروتوكول توجيه (OSPF، EIGRP) على كل وصلة من قفزة إلى قفزة إلى قيمة موجبة صغيرة.
- حدد بروتوكول التوجيه EIGRP إذا كان نوع البيانات ملفات نصية.
- حدد بروتوكول التوجيه OSPF إذا كان نوع البيانات فيديو.
- حدد بروتوكول التوجيه EIGRP إذا كان نوع البيانات صوت.

إنشاء النمل والوصول للوجهة وفق مراحل دورة حياة النمل بالخوارزمية المطورة:

1– المرحلة الأولى من دورة حياة النمل:

- إنشاء نملة جديدة
- تبدأ كل نملة من عقدة المصدر.

- تسير النملة عشوائياً عبر الشبكة بحثاً عن عميل متوافق مع نوع البيانات (Video، Voice، Text).
- تستخرج النملة خلال رحلتها نوع البيانات من الحزم المارة في الشبكة.
- تستخدم النملة مزيجاً من ACO و DFS لاستكشاف شامل للمسارات الممكنة مع القيود التالية:
  - يتم إزالة أي انحياز احتمالي في اختيار العقد (جعل جميع الخيارات متساوية الاحتمال).
  - يتم تعطيل تبخر الفيرمون، للحفاظ على جميع المسارات المكتشفة.
  - لا يُسمح للنملة بزيارة نفس العقدة أكثر من مرتين (لمنع الحلقات).
  - لا يُسمح للنملة بالتوجه إلى عميل غير متوافق
  - إذا وصلت لطريق مسدود تستخدم منطق الرجوع للخلف (backtracking)
- التحقق من صحة الهدف
- تحفظ النملة المسارات الكاملة فقط، ويُطبق حد أقصى لطول المسار (30 عقدة).
- يتم تصنيف كل مسار حسب نوع الخدمة (نص/صوت/فيديو).
- يتم استبعاد أي مسار يصل إلى عميل غير تابع لنوع الخدمة الصحيح.
- تجاوز الوصلات الفاشلة
- توازن الحمولة عبر اختيار الجيران الأقل ازدحاماً
- انتقال النملة إلى مرحلة العودة

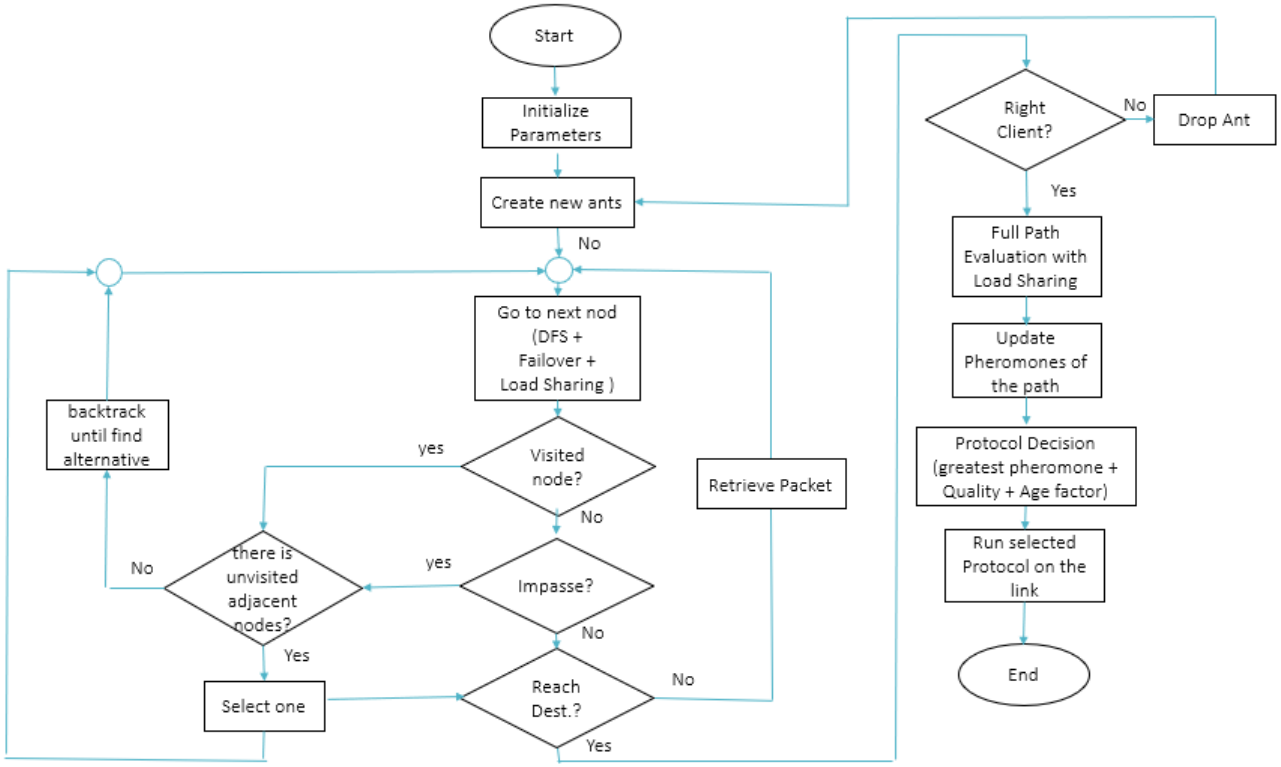
2- المرحلة الثانية من دورة حياة النمل:

- عند الوصول إلى العقدة الهدف (العميل الصحيح المتوافق مع نوع السيرفر)، تعود النملة إلى المصدر.
- أثناء العودة (مرحلة RETURN):
  - يتم تسجيل المسار الكامل إلى العميل.
  - يتم عكس المسار (reverse) لتبدأ العودة إلى المصدر.
  - في طريق العودة، تستمر النملة بتسجيل الخصائص من كل عقدة لتجمع بيانات المسار كاملة (تأخير، عرض الحزمة، التذبذب، الفقد)
  - يتم تقييم المسار بالكامل (حسب التأخير، الطول، نوع البروتوكول المناسب ....).
  - حساب الفيرمون النهائي التراكمي للبروتوكولين (OSPF و EIGRP) على كامل المسار بناءً على هذا التقييم.

3- المرحلة الثالثة من دورة حياة النمل:

- النملة تنطلق من المصدر مرة أخرى، عبر نفس المسار، ولكن بهدف إخبار كل راوتر على المسار بالبروتوكول المختار.
- كل راوتر في هذا المسار يفعّل البروتوكول المناسب حسب قرار النملة.
- عند الوصول إلى العميل، تختفي النملة.

1.9.7. مخطط الخوارزمية المقترحة النهائية:



الشكل (7-32)- مخطط الخوارزمية المقترحة بعد التطوير الثاني

الفصل الثامن

الإسهامات العلمية والأفق المستقبلية

Scientific Contributions and Future Perspectives

1.8. خلاصة البحث :

هذه الخوارزمية تقدّم نموذجاً مطوراً من خوارزمية مستعمرة النمل الكلاسيكية (ACO) أُعيد تصميمها لتتلاءم مع بيئة شبكات متعددة الخدمات (Multiservice Networks) عبر خوارزمية أطلقنا عليها اسم : *ACO-Driven Protocol Selection (**ADPS**)* . هذا النموذج يدمج عدة أنواع من التعديلات المعتمدة عالمياً، بما في ذلك على سبيل المثال:

- **التعديل التخصصي** لتكييف ACO لمهمة التوجيه الذكي بين بروتوكولات OSPF و EIGRP حسب نوع البيانات.
- **التعديل الهيكلي الهجين** عبر دمج ACO مع خوارزمية DFS لتحسين اكتشاف المسارات، و تسجيل كامل المسارات داخل ذاكرة النملة وعدم الاكتفاء بمسار واحد.
- **تعديلات قائمة على قيود جودة الخدمة (QoS)** مثل: jitter و loss و delay و bandwidth.
- **تعديل قائم على توافق نوع الخدمة** عبر تصفية المسارات غير المتوافقة بين نوع السيرفر والعميل.
- **تحقق ديناميكي متعدد الطبقات** لضمان استقرار المسار وتوافقه مع نوع الخدمة.
- **آلية التحمل والانهيار** من خلال اكتشاف مسارات متعددة يمكن الاعتماد عليها عند تعطل أحد الوصلات، مما يعزز من مرونة الشبكة واستمرارية الخدمة.
- **موازنة التحميل عبر المسارات المتعددة** من أجل تحسين توزيع استكشاف المسارات وتجنب ازدحام مبكر في الشبكة.

والنتيجة هي نموذج توجيه جديد يتفوق على ACO التقليدية من حيث دقة اختيار البروتوكول وكفاءة الأداء الشبكي، والتكيف مع البيئات المركّبة والمتغايرة الخدمة، مما يجعله أداة فعالة لمحاكاة التوجيه الذكي في البيئات المركّبة.

وعلى الرغم من الانتشار الواسع لخوارزمية مستعمرة النمل ACO في مجال التوجيه الشبكي، إلا أن الاستخدام التقليدي لها غالباً ما اقتصر على اكتشاف المسار الأفضل بناءً على مقاييس ثابتة مثل الطول أو التكلفة، دون النظر إلى التكيف الديناميكي مع طبيعة الخدمات الشبكية أو بنية البروتوكولات. كما أنه لم يتم في السابق دمج هيكلي للخوارزمية ACO مع خوارزميات بحث مثل DFS، أو تغيير في وظيفتها ليشمل اختيار بروتوكول التوجيه ضمن عملية اتخاذ القرار لدى كل نملة بالإضافة لمراعاة قيود جودة الخدمة، لذلك فقد قدم هذا البحث نموذجاً هجيناً ومتعدد الأبعاد لخوارزمية ACO، حيث تميز بدمج متكامل لمجموعة من التعديلات الهيكلية والسلوكية والوظيفية.

وبناءً على مراجعة الأدبيات، لم تُسجَّل أي دراسة سابقة جمعت هذا العدد من التعديلات في نموذج توجيهي واحد قائم على خوارزمية ACO ضمن بيئة شبكات متعددة الخدمات، مما يجعل هذا النموذج مساهمة نوعية في مجال التوجيه الذكي القابل للتكيف.

2.8. الإسهامات العلمية للبحث

يمثل هذا البحث إضافة نوعية إلى ميدان خوارزميات التوجيه في الشبكات من خلال الدمج المبتكر بين منهجيات الذكاء المستوحى من الطبيعة (خوارزمية مستعمرة النمل ACO) وبروتوكولات التوجيه التقليدية (OSPF وEIGRP). ويمكن تلخيص الإسهامات العلمية الرئيسة لهذا العمل فيما يلي:

تطوير إطار هجين للتوجيه: قدم البحث خوارزمية جديدة تمزج بين قدرات الاستكشاف العشوائي المنظم في ACO وبين قوة بروتوكولات التوجيه القياسية، مما أتاح آلية مرنة وديناميكية لاختيار المسارات المثلى.

دعم مبدأ تعددية المسارات (Multipath Routing): يساهم النموذج المقترح في موازنة الأحمال بالمسارات المكتشفة ويوفر قدرة تجاوز سريعة عند الأعطال (Failover).

تكامل التوجيه مع أنواع متعددة من الخدمات: من خلال تخصيص المسارات بناءً على نوع البيانات (فيديو، صوت، نص)، يعكس البحث بعداً جديداً في مجال التوجيه القائم على جودة الخدمة (QoS-aware Routing)، ويؤسس لنهج يوازن بين متطلبات التطبيقات الحساسة للزمن واحتياجات البيانات التقليدية.

إرساء منهجية تقييم مقارنة: قدّم البحث إطاراً متكاملاً لتقييم الأداء عبر ثلاثة سيناريوهات مستقلة (الخوارزمية المقترحة، بروتوكول OSPF، وبروتوكول EIGRP)، مما أتاح إجراء مقارنات كمية دقيقة وفق معايير متعددة تشمل التأخير، التذبذب، فقد الحزم، وزمن الاستجابة.

إسهام في ربط النظرية بالتطبيق: يقدّم البحث أداة عملية قابلة لإعادة الاستخدام والتوسيع في دراسات لاحقة، ويسهّل نقل الخوارزمية من بيئة المحاكاة إلى تطبيقات عملية في شبكات حقيقية.

بذلك، لا يقتصر إسهام البحث على تحسين أداء التوجيه والإدارة في الشبكات فقط، بل يمتد ليشكل قاعدة بحثية يمكن البناء عليها مستقبلاً لتطوير بروتوكولات هجينة أكثر ذكاءً وقابلية للتكيف مع متطلبات الشبكات الحديثة.

3.8. التطبيقات المقترحة للبحث

يمكن النظر إلى الخوارزمية المقترحة المطورة (ADPS) كأداة فعّالة لتحسين أداء الشبكات التي تتميز بالتعقيد وكثرة التدفقات، لما توفره من قدرة على استكشاف المسارات المتعددة وتوزيع الموارد بشكل ديناميكي، وضمان استمرارية الخدمة عبر خاصية الـ Failover، حيث يُنصحها لخصائصها التالية:

- قابلة للتوزيع وذات اعتماد منخفض على عنصر مركزي فهي مناسبة للشبكات الديناميكية.
- تكتشف مسارات متعددة تلقائياً، ما يعزز موازنة الحمل ويدعم Failover دون تدخل يدوي.
- دعم صريح لتمييز الخدمات عبر فيرمونات مخصّصة لكل نوع بيانات، ما يجعلها مناسبة لأنظمة QoS متعددة الطبقات.
- مرونة معايير القياس: يمكن تضمين متغيرات إضافية (طاقة، أمن، تكلفة) في دالة الجودة بسهولة.

ومن أبرز التطبيقات التي يمكن أن تستفيد من هذه الخوارزمية:

- **الشبكات المؤسسية (Enterprise Networks):**

يمكن استخدام الخوارزمية داخل شبكات المؤسسات الكبرى التي تعتمد على خدمات متعددة (الفيديو، الصوت، البيانات)، حيث تساهم في ضمان جودة الخدمة لكل نوع بيانات عبر اختيار المسارات الأمثل وتفعيل مبدأ موازنة الأحمال.

- **شبكات البنية التحتية الحرجة (Critical Infrastructure Networks):**

مثل الشبكات المالية أو الصحية، حيث يُعتبر استقرار الخدمة واستمراريتها أمراً ضرورياً. تدعم الخوارزمية تجاوز الأعطال (Failover) وتوفّر مسارات بديلة فورية تضمن الحفاظ على الاتصال حتى في حال حدوث انقطاعات.

- **شبكات الجامعات والمراكز البحثية:**

هذه البيئات تتطلب توجيهاً فعالاً بسبب حجم البيانات الكبير الناتج عن تطبيقات البحث العلمي والمحاكاة. توفر الخوارزمية مسارات متوازية تساهم في تقليل التأخير وتحسين زمن الاستجابة لتطبيقات التحليل والاختبار.

- **شبكات ومراكز القيادة والسيطرة:**

في البيئات التي تحتاج إلى موثوقية عالية وأمان متزايد، يمكن للخوارزمية أن تعمل على تعزيز استقرار الاتصال وضمان استمرارية الخدمات الحرجة مع الحفاظ على جودة الأداء.

- **الشبكات المعرفة بالبرمجيات SDN (Software-Defined Networking) وأنظمة إدارة الشبكات الذكية:**

يمكن دمج الخوارزمية ضمن منظومات التحكم المركزي لتكون طبقة ذكية لإدارة المسارات المتعددة واختيار بروتوكولات التوجيه المناسبة، مع إمكانية تعديل سياسات التوجيه ديناميكياً بما يتوافق مع احتياجات التطبيقات والخدمات.

4.8. التوصيات والآفاق المستقبلية:

تُعدّ الخوارزمية المقترحة المطورة (ADPS) التي تمزج بين الخوارزميات (ACO) و (DFS) وبروتوكولات التوجيه التقليدية مثل OSPF و EIGRP إطاراً واعداً لتطوير أنظمة توجيه ذكية ومرنة، حيث أظهرت قدرة واضحة على استكشاف المسارات المتعددة وتطبيق استراتيجيات موازنة الأحمال والتجاوز عند الأعطال. ومن أجل تعزيز القيمة العلمية والعملية لهذه الخوارزمية، يمكن توظيف الخوارزمية مستقبلاً في البيئات اللاسلكية وشبكات إنترنت الأشياء (IoT)، إذ يتيح مبدأ الاستكشاف التعاوني للنمل استيعاب التغيرات الديناميكية للطوبولوجيا والتعامل بكفاءة مع شبكات كثيفة العقد. وهذا يتطابق تماماً مع متطلبات الشبكات اللاسلكية وبيئات إنترنت الأشياء: طوبولوجيا ديناميكية، تغيّر وصلات متكرر، وحاجة إلى توجيه موزّع لا يعتمد على نقطة مركزية. في مثل هذه البيئات، يمكن الاعتماد على فيرمونات مخصصة لكل نوع خدمة لضمان استمرارية الاتصال وتحقيق كفاءة عالية في استهلاك الموارد.

Refrences المراجع

1. Monita V., Irawati I., Tulloh R. (2020). Comparison of routing protocol performance on multimedia services on software defined network, *Bulletin of Electrical Engineering and Informatics, published by the Institute of Advanced Engineering and Science (IAES),* Vol. 9, No. 4, ISSN: 2302-9285, DOI: 10.11591/eei.v9i4.2389.

2. S.Anusuya, S.Baulkani, (2022) Performance Analysis of Routing Protocols RIP, OSPF and EIGRP, *International Research Journal of Engineering and Technology (IRJET),* Volume: 09 Issue: 06, p-ISSN: 2395-0072

3. Hossain A., Ali M, Akter S. (2020). Performance Comparison of EIGRP, OSPF and RIP Routing Protocols using Cisco Packet Tracer and OPNET Simulator, *Global Journal of Computer Science and Technology*, DOI: 10.34257 / GJCSTGVOL20IS2PG1, Volume 20 Issue 2 Version 1.0, Online ISSN: 0975-4172 & Print ISSN: 0975-4350**.**

4. Simson, F and Widiasari, I. (2023), Comparative analysis of quality of service performance of video streaming services using OSPF and EIGRP networks, JIKO (Jurnal Informatika dan Komputer), Vol. 6, No. 1, April 2023, hlm. 36-42, DOI: 10.33387/jiko.v6i1.5826.

5. Biradar A. (2020) Comparative Study on Routing Protocols: RIP, OSPF and EIGRP and Their Analysis Using GNS-3. IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), doi: 10.1109/ICRAIE51050.2020.9358327

6. Tsochev, G., Popova K. and Stankov I. (2021), A comparative study by simulation of OSPF and EIGRP routing protocols, Informatics and Automation. Vol. 21 No. 6. ISSN 2713-3192 (print) 1241, ISSN 2713-3206 (online) www.ia.spcras.ru, DOI 10.15622/ia.21.6.6.

7. Noelia, K. Dimitrios K. (2023), A Comparative Analysis of OSPF and EIGRP Routing Protocol Evaluation, Journal of Transactions in Systems Engineering, DOI: 10.15157/JTSE.2023.1.2.73-103

8. Shahid K, Ahmad S, Rizvi S. (2024), Optimizing Network Performance: A Comparative Analysis of EIGRP, OSPF, and BGP in IPv6-Based Load-Sharing and Link-Failover Systems, Future Internet MDPI, Volume 16, Issue 9, :DOI: 10.3390/fi16090339.

9. Ameen O, Akande B and Kehinde A. (2016) Solving Network Routing Problem Using Artificial Intelligent Techniques, British Journal of Mathematics & Computer Science. DOI:10.9734/BJMCS/2016/27033.

10. Mamta, Pooja A. (2016) ANTS-BASED ALGORITHMS TO ROUTING PROTOCOLS, International Journal of Engineering Applied Sciences and Technology, Vol. 1, Issue 8, ISSN No. 2455-2143, Pages 153-156

11. Marco D, Mauro B, Thomas S. (2006) Ant colony optimization, Published in IEEE Computational, Universite´ Libre de Bruxelles, BELGIUM, Computer Science.

12. Dahan F, Hindi K, Ghoneim A,and Alsalman H. (2017) An Enhanced Ant Colony Optimization Based Algorithm to Solve QoS-Aware Web Service Composition, DOI 10.1109/ACCESS.2021.3061738, IEEE Access.

13. Cormen, Thomas H., et al. (2009) Introduction to Algorithms. 3rd ed., MIT Press, DOI: 9435.001.0001

14. Steve McQuerry, (2008) Interconnecting Cisco Network Devices, Part 2 (ICND2), Cisco Press, CCIE No. 6108.

15. Usuari, (2010) Comparative analysis of the routing protocols RIPv2, OSPFv2 and Integrated IS-IS, Opnet & La Salle Reports.

16. Dordal P. (2017). An Introduction to Computer Networks, Edition 1.9.3, *Department of Computer Science, Loyola University Chicago*, 771.

17. Bertsekas D., Gallager R. (1992) Data Networks (2nd Edition), Massachusetts Institute of Technology, ISBN-13: 978-1-886529-22-9

18. Ramakrishnan R., Gehrke J., Derstadt J. (2022) Database Management Systems 3rd Edition, University of Wisconsin, Published by McGraw-Hili, ISBN: 0072465638.

19. Forouzan B. (2012), *Data Communications and Networking 5th ed*, Published by McGraw-Hill, ISBN 978-0-07-337622-6.

20. Sachinidis T., Politis AC., Hilas CS. (2023) *To Split or not to Split? A Simulation Study on the Network Convergence Duration of Multi-Area OSPF*, Publisher: IEEE, DOI: 10.1109/TSP59544.2023.10197672.

21. Akintola G., Falade P. (2024) *Evaluating the Performances of Digital Data Transmission Links on a Client-Server Network Model*, International Journal of Scientific Research in Computer Science and Engineering, Vol.12, Issue.2, pp.53-66, E-ISSN: 2320-7639

22. Kristianto P., Putra A. (2021) *Comparative Analysis of IPv4 and IPv6 OpenVPN Protocol Performance Based on QoS Parameters*, Journal of Advances in Information Systems and Technology, DOI: 10.15294/jaist. v3i1.49095

23. Sommers J. Barford P. Duffield N. Ron A. (2005) Improving Accuracy in End-to-end Packet Loss Measurement, *Philadelphia, USA. Copyright ACM SIGCOMM* Computer Communication Review Volume 35 Issue 4.

24. Zhang T., Liu B. (2019) Exposing End-to-End Delay in Software-Defined Networking, *International Journal of Reconfigurable Computing*, ID 7363901 | https://doi.org/10.1155/2019/7363901

25. Wee S., Tan W., Apostolopoulos J. Etoh M. (2002), OPTIMIZED VIDEO STREAMING FOR NETWORKS WITH VARYING DELAY, *Conference Paper, Source: IEEE Xplore*, DOI: 10.1109/ICME.2002.1035431.

26. Moudi M., Othman M. (2020), On the relation between network throughput and delay curves, *Journal for Control, Measurement, Electronics, Computing and Communications* Volume 61, Issue 3, https://doi.org/10.1080/00051144.2020.1774731.

27. Jianli X, Binhan Z, Cuiran L (2023), Research of 5G HUDN network selection algorithm based on Dueling-DDQN, EURASIP Journal on Wireless Communications and Networking, Article number: 113.

28. Sylvia J. et al.(2011), The Measurement and Analysis of Housing Preference and Choice, Springer Science Business Media B.V., DOI 10.1007/978-90-481-8894-9_5.

29. G.1010 : End-user multimedia QoS categories, ITU: Committed to connecting the world, https://www.itu.int/rec/t-rec-g.1010-200111-i

30. Yang X (2020), Nature-Inspired Computation and Swarm Intelligence Algorithms, Theory and Applications. 1st Edition -April 9, 2020, published by Elsevier Ltd, London, ISBN: 9780128197141.

31. Cai W, Jin X, Zhang Y. (2006), ACO-Based QoS Routing Algorithm for Wireless Sensor Networks, Springer – Lecture Notes in Computer Science, Vol. 4159.

الملحق

(كود الخوارزمية الأساسي)

```
import random
import time
import networkx as nx
import matplotlib.pyplot as plt
from collections import defaultdict, deque
from enum import Enum
from dataclasses import dataclass, field
from typing import Set, Dict, List, Deque, Optional, Tuple,
DefaultDict
import numpy as np
import re
import heapq
from collections import defaultdict, deque
NETWORK_SAMPLES = []
def ospf_routing(network_graph, source, destination):
    if not isinstance(network_graph, dict) or not
network_graph:
        raise ValueError("يجب أن يكون رسم الشبكة عبارة عن قاموس غير فارغ")

    if source not in network_graph:
        raise ValueError(f" العقدة المصدر{source} غير موجودة في الشبكة")

    if destination not in network_graph:
        raise ValueError(f" العقدة الهدف{destination} غير موجودة في
الشبكة")

    # تهيئة هياكل البيانات
    distances = defaultdict(lambda: float('infinity'))
```

```
    distances[source] = 0
    previous_nodes = {}
    priority_queue = [(0, source)]
    visited = set()
    while priority_queue:
        current_distance, current_node =
heapq.heappop(priority_queue)

        if current_node == destination:
            break

        if current_node in visited:
            continue
        visited.add(current_node)

        for neighbor, link_data in
network_graph.get(current_node, {}).items():
            cost = link_data['ospf_cost']
            distance = current_distance + cost
            if distance < distances[neighbor]:
                distances[neighbor] = distance
                previous_nodes[neighbor] = current_node
                heapq.heappush(priority_queue, (distance,
neighbor))

    # بناء المسار إذا كان موجوداً
    if destination not in previous_nodes and source !=
destination:
        return None, float('infinity'), {}

    path = []
    current_node = destination
    while current_node is not None:
```

```
        path.append(current_node)
        current_node = previous_nodes.get(current_node)
    path.reverse()

    if not path or path[0] != source:
        return None, float('infinity'), {}

    return path, distances[destination], {}

def eigrp_routing(network_graph, source, destination, k1=1,
k2=0, k3=1, k4=0, k5=0):

    if not isinstance(network_graph, dict) or not
network_graph:
        raise ValueError("يجب أن يكون رسم الشبكة عبارة عن قاموس غير فارغ")

    if source not in network_graph:
        raise ValueError(f"العقدة المصدر {source} غير موجودة في الشبكة")

    if destination not in network_graph:
        raise ValueError(f"العقدة الهدف {destination} غير موجودة في
الشبكة")

    def calculate_metric(metrics):
        bandwidth, delay, reliability, load, mtu = metrics
        try:
            bw_component = (10**7 / bandwidth) if bandwidth >
0 else float('inf')
            delay_component = delay / 10

            reliability_component = 1.0
            if k5 != 0:
```

```
                reliability_component = k5 / (reliability +
k4) if (reliability + k4) != 0 else float('inf')

            load_component = 0.0
            if k2 != 0 and load < 256:
                load_component = k2 * bw_component / (256 -
load)

            composite = (k1 * bw_component + load_component +
k3 * delay_component) * reliability_component
            return composite
        except (TypeError, ValueError, ZeroDivisionError):
            return float('inf')

    weighted_graph = defaultdict(dict)
    for node in network_graph:
        for neighbor, metrics in network_graph[node].items():
            weight =
calculate_metric(metrics['eigrp_metrics'])
            weighted_graph[node][neighbor] = weight

    distances = defaultdict(lambda: float('infinity'))
    distances[source] = 0
    previous_nodes = {}
    priority_queue = []
    heapq.heappush(priority_queue, (0, source))

    while priority_queue:
        current_distance, current_node =
heapq.heappop(priority_queue)

        if current_node == destination:
            break
```

```
        if current_distance > distances[current_node]:
            continue

        for neighbor, weight in
weighted_graph.get(current_node, {}).items():
            new_distance = current_distance + weight

            if new_distance < distances[neighbor]:
                distances[neighbor] = new_distance
                previous_nodes[neighbor] = current_node
                heapq.heappush(priority_queue, (new_distance,
neighbor))

    if destination not in previous_nodes and source !=
destination:
        return None, float('infinity'), {}

    path = []
    current_node = destination
    while current_node is not None:
        path.append(current_node)
        current_node = previous_nodes.get(current_node)
    path.reverse()

    if not path or path[0] != source:
        return None, float('infinity'), {}

    return path, distances[destination], {}


class DataType(Enum):
    TEXT = 0
    VOICE = 1
    VIDEO = 2
```

```
class Protocol(Enum):
    OSPF = "OSPF"
    EIGRP = "EIGRP"
    UNDEFINED = "UNDEFINED"

class AntPhase(Enum):
    EXPLORATION = 1
    RETURN = 2
    DECISION = 3

class NodeType(Enum):
    SERVER = "server"
    ROUTER = "router"
    CLIENT = "client"

@dataclass
class NetworkConditions:
    delay: float
    bandwidth: float
    loss: float
    jitter: float
    timestamp: float = field(default_factory=time.time)

@dataclass
class HopRecord:
    node_id: str
    arrival_time: float
    departure_time: float
    conditions: NetworkConditions
    ospf_before: float
    eigrp_before: float
    ospf_after: float
    eigrp_after: float
```

```
    processing_delay: float

@dataclass
class Ant:
    id: str
    source: str
    destination: str
    data_type: DataType
    source_mark: bool = True
    payload: Optional[bytes] = None
    network: Optional['Network'] = None
    current_node: str = field(init=False)
    forward_path: List[str] = field(default_factory=list)
    return_path: List[str] = field(default_factory=list)
    full_path: List[str] = field(default_factory=list)
    path: Deque[str] = field(default_factory=deque)
    ospf_pheromone: float = 1.0
    eigrp_pheromone: float = 1.0
    phase: AntPhase = AntPhase.EXPLORATION
    protocol_decision: Protocol = Protocol.UNDEFINED
    creation_time: float = field(default_factory=time.time)
    hops: int = 0
    total_delay: float = 0.0
    hop_history: List[HopRecord] = field(default_factory=list)
    status: str = "ACTIVE"
    bandwidth_samples: List[float] =
field(default_factory=list)
    delay_samples: List[float] = field(default_factory=list)
    loss_samples: List[float] = field(default_factory=list)
    jitter_samples: List[float] = field(default_factory=list)
    all_paths: Dict[str, List[List[str]]] =
field(default_factory=lambda: defaultdict(list))
    max_paths_per_destination: int = 10
    max_path_length: int = 30
```

```
    current_path: List[str] = field(default_factory=list)
    visited_nodes: Set[str] = field(default_factory=set)
    timeout: float = 60.0
    visited_times: Dict[str, int] =
field(default_factory=lambda: defaultdict(int))
    dfs_stack: List = field(default_factory=list)
    qos_samples: List[Dict[str, float]] =
field(default_factory=list)


    def is_expired(self):
        return time.time() - self.creation_time > self.timeout

    def __post_init__(self):
        self.current_node = self.source
        self.path = deque([self.source])
        self.current_path = [self.source]
        self.visited_nodes.add(self.source)
        self.max_paths_per_destination = 10  # الحد الأقصى للمسارات
لكل هدف
        self.max_path_length = 30  # الحد الأقصى لطول المسار
        self.path_history = []  # لتسجيل تاريخ المسار بالكامل
        self.visited_times = defaultdict(int)  # عدد مرات زيارة كل
عقدة
        self.dead_nodes = set()  # عقد تم الرجوع منها بالكامل ويمنع زيارتها مجدداً


    def get_processing_delay(self) -> float:
        return self.base_processing_delay * (1 +
self.current_load)
```

```
    def record_hop(self, node_id: str, conditions:
NetworkConditions, processing_delay: float):
        # تسجيل القفزة دون تحديث فيرمونات مؤقتة
        if self.phase == AntPhase.EXPLORATION and node_id not
in self.forward_path:
            self.forward_path.append(node_id)
        elif self.phase == AntPhase.RETURN and node_id not in
self.return_path:
            self.return_path.append(node_id)

        self.path.append(node_id)
        self.current_node = node_id
        self.hops += 1
        self.total_delay += processing_delay

        self.hop_history.append(HopRecord(
            node_id=node_id,
            arrival_time=time.time(),
            departure_time=time.time() + processing_delay,
            conditions=conditions,
            ospf_before=0.0,
            eigrp_before=0.0,
            ospf_after=0.0,
            eigrp_after=0.0,
            processing_delay=processing_delay
        ))

    def is_node_compatible(self, node_id: str, network:
'Network') -> bool:
        """
        تفحص إن كانت العقدة مناسبة لنوع بيانات النملة.
        """
```

```
        node_data_type =
network.graph.nodes[node_id].get("data_type")
        if node_data_type is None:
            return True  # عقدة محايدة
        return node_data_type == self.data_type

    def record_path(self, destination: str):
        """تسجيل المسار فقط إذا كان متوافقاً مع نوع الخدمة"""
        # التحقق من توافق نوع الخدمة
        if 'Video' in self.source:
            if not ('Video_Client' in destination or
'Mixed_Client' in destination):
                print(f"❌ Path recording blocked: Video
server to non-video client")
                return False
        if ('Voice' in self.source or 'DB' in self.source) and
'Video_Client' in destination:
            print(f"❌ Path recording blocked: Non-video
server to video client")
            return False

        path_str = '->'.join(self.current_path)
        if len(self.current_path) > self.max_path_length:
            return False

        # منع تكرار المسار
        for existing_path in self.all_paths[destination]:
            if path_str == '->'.join(existing_path):
                return False
```

```
        if len(self.all_paths[destination]) >=
self.max_paths_per_destination:
            return False

        self.all_paths[destination].append(self.current_path.c
opy())
        return True



    def get_unvisited_neighbors(self, network: 'Network') ->
List[str]:
        nbrs =
list(network.graph.neighbors(self.current_node))
        valid = []
        for n in nbrs:
            if self.visited_times[n] >= 2:
                continue
            if not self.is_node_compatible(n, network):
                continue
            valid.append(n)
        random.shuffle(valid)
        return valid

    def move_to_node(self, node: str):

        if not self.is_node_compatible(node, network):
            print(f"🚫 Ant {self.id} blocked from moving to
{node} due to incompatibility")
            self.status = "FAILED"
            return
        """انتقال النملة إلى عقدة جديدة"""
        self.visited_nodes.add(node)
        self.current_path.append(node)
```

```
        self.current_node = node
        self.path.append(node)
        self.path_history.append(node)
        self.visited_times[node] += 1
        # اكتشاف الحلقات
        if self.visited_times[node] > 2:  # إذا زار العقدة أكثر من مرتين
            print(f"🔄 Ant {self.id} detected loop at {node}")
            self._handle_loop(node)

    def _handle_loop(self, node: str):
        """عند اكتشاف حلقة، نحذف العقدة الحالية ونعود لعقدة سابقة"""
        if len(self.current_path) > 1:
            self.current_path.pop()
            self.current_node = self.current_path[-1]
        else:
            self.status = "FAILED"

    def get_next_possible_nodes(self, current_node: str,
network: 'Network') -> List[str]:
        """الحصول على العقد التالية الممكنة بدون انحياز احتمالي"""
        possible_nodes = []
        for neighbor in network.nodes[current_node].neighbors:
            if neighbor not in self.visited_nodes:
                possible_nodes.append(neighbor)

        # إذا لم يكن هناك عقد متاحة، حاول العودة للخلف
        if not possible_nodes and len(self.current_path) > 1:
            self.current_path.pop()
            prev_node = self.current_path[-1]
            possible_nodes = [prev_node]
```

```
        return possible_nodes


    @staticmethod
    def normalize(value, min_val, max_val):
        return (value - min_val) / (max_val - min_val + 1e-6)

    def update_pheromones(self, conditions:
NetworkConditions):
        import numpy as np

        Qs = self.calculate_quality_factor(conditions)  #
normalized quality

        q_set = getattr(self, 'q_set', 0.6)
        dead  = getattr(self, 'q_deadband', 0.05)
        HN    = getattr(self, 'hysteresis_N', 2)

        err = q_set - Qs  # +ve => below target
        if not hasattr(self, '_below_cnt'):
            self._below_cnt = 0
        if not hasattr(self, '_above_cnt'):
            self._above_cnt = 0
        if err >  dead:
            self._below_cnt += 1; self._above_cnt = 0
        elif err < -dead:
            self._above_cnt += 1; self._below_cnt = 0
        else:
            self._below_cnt = 0; self._above_cnt = 0

        gain_below = getattr(self, 'gain_below', 2.0)  # push
when below band
        gain_above = getattr(self, 'gain_above', 1.0)  # damp
when above band
```

```
        mult = 1.0
        if self._below_cnt >= HN:
            mult *= (1.0 + gain_below * (err - dead))
        elif self._above_cnt >= HN:
            mult *= max(0.5, 1.0 - gain_above * ((-err) -
dead))

        if not hasattr(self, '_Q_hist'):
            self._Q_hist = []
        self._Q_hist.append(float(Qs))
        WINDOW_Q = getattr(self, 'delta_auto_window', 200)
        if len(self._Q_hist) > WINDOW_Q:
            self._Q_hist = self._Q_hist[-WINDOW_Q:]

        if self._Q_hist:
            Q_mean = float(np.mean(self._Q_hist))
            Q_std  = float(np.std(self._Q_hist))
        else:
            Q_mean, Q_std = 0.5, 0.1

        alpha_min   = getattr(self, 'delta_auto_alpha_min',
0.30)  # قاعدة لمستوى Δmin
        gamma_floor = getattr(self, 'delta_auto_gamma_floor',
0.10)
        beta_max    = getattr(self,
'delta_auto_beta_max',  1.60)
        lambda_span = getattr(self, 'delta_auto_lambda_span',
1.50)

        dmin_lo, dmin_hi = 0.10, 2.00
        dmax_lo, dmax_hi = 0.60, 3.50
```

```
        dmin = alpha_min * (1.0 - Q_mean) + gamma_floor *
Q_std
        dmax = beta_max  * (1.0 + lambda_span * Q_std)

        dmin = float(np.clip(dmin, dmin_lo, dmin_hi))
        dmax = float(np.clip(dmax, dmax_lo, dmax_hi))

        if dmax <= dmin + 1e-6:
            dmax = min(dmin + 0.40, dmax_hi)

        span = dmax - dmin

        base_delta = dmin + span * Qs
        delta = base_delta * mult

        DataType_ = globals().get('DataType', None)
        svc = getattr(self, 'data_type', None)
        if svc is None and hasattr(self, 'ants') and
getattr(self, 'ants', None):
            svc = getattr(self.ants[0], 'data_type', None)
        if svc is None and DataType_ is not None:
            svc = DataType_.VIDEO
        name = str(getattr(svc, 'name', svc)).upper()

        cap = getattr(self, 'pheromone_cap', 1e6)
        if name == 'VIDEO':
            self.ospf_pheromone = min(max(getattr(self,
'ospf_pheromone', 0.0) + delta, 0.0), cap)
            print(f"🤖 OSPF pheromone += {delta:.2f} →
{self.ospf_pheromone:.2f}  "
                  f"[dmin={dmin:.2f}, dmax={dmax:.2f},
meanQ={Q_mean:.2f}, stdQ={Q_std:.2f}]")
        elif name in ('VOICE', 'TEXT'):
```

```
            self.eigrp_pheromone = min(max(getattr(self,
'eigrp_pheromone', 0.0) + delta, 0.0), cap)
            print(f"🎧/📝 EIGRP pheromone += {delta:.2f} →
{self.eigrp_pheromone:.2f}  "
                  f"[dmin={dmin:.2f}, dmax={dmax:.2f},
meanQ={Q_mean:.2f}, stdQ={Q_std:.2f}]")
        else:
            print(f"⚠ Unknown service type '{name}',
delta={delta:.2f} (no pheromone updated)  "
                  f"[dmin={dmin:.2f}, dmax={dmax:.2f},
meanQ={Q_mean:.2f}, stdQ={Q_std:.2f}]")

        if not hasattr(self, 'k_params') or not
isinstance(self.k_params, dict):
            self.k_params = {'bandwidth': 0.6, 'delay': 0.6,
'jitter': 0.6, 'loss': 0.6}

        k_min   = getattr(self, 'k_min', 0.20)
        k_max   = getattr(self, 'k_max', 0.90)
        eta     = getattr(self, 'k_eta', 0.05)
        alpha_k = getattr(self, 'k_ema_alpha', 0.20)

        if not hasattr(self, '_ema_bw'):
            self._ema_bw = conditions.bandwidth
        else:
            self._ema_bw = (1 - alpha_k) * self._ema_bw +
alpha_k * conditions.bandwidth

        if not hasattr(self, '_ema_d'):
            self._ema_d = conditions.delay
        else:
            self._ema_d = (1 - alpha_k) * self._ema_d +
alpha_k * conditions.delay
```

```
        if not hasattr(self, '_ema_j'):
            self._ema_j = conditions.jitter
        else:
            self._ema_j = (1 - alpha_k) * self._ema_j +
alpha_k * conditions.jitter

        if not hasattr(self, '_ema_l'):
            self._ema_l = conditions.loss
        else:
            self._ema_l = (1 - alpha_k) * self._ema_l +
alpha_k * conditions.loss

        if name == 'VIDEO':
            bw_cap = 4000.0
        elif name == 'VOICE':
            bw_cap = 100.0
        else:
            bw_cap = 50.0

        bw_target = 0.60 * bw_cap
        d_target  = 0.150
        j_target  = 0.020
        l_target  = 0.010

        e_bw = max(0.0, (bw_target - self._ema_bw) /
(bw_target + 1e-12))
        e_d  = max(0.0, (self._ema_d - d_target) / (d_target +
1e-12))
        e_j  = max(0.0, (self._ema_j - j_target) / (j_target +
1e-12))
        e_l  = max(0.0, (self._ema_l - l_target) / (l_target +
1e-12))
```

```
        kb = float(np.clip(self.k_params['bandwidth'] + eta *
e_bw, k_min, k_max))
        kd = float(np.clip(self.k_params['delay']     + eta *
e_d,  k_min, k_max))
        kj = float(np.clip(self.k_params['jitter']    + eta *
e_j,  k_min, k_max))
        kl = float(np.clip(self.k_params['loss']      + eta *
e_l,  k_min, k_max))

        beta = getattr(self, 'k_momentum', 0.0)
        if beta > 0.0 and hasattr(self, 'k_params_prev'):
            kb = beta * self.k_params['bandwidth'] + (1 -
beta) * kb
            kd = beta * self.k_params['delay']     + (1 -
beta) * kd
            kj = beta * self.k_params['jitter']    + (1 -
beta) * kj
            kl = beta * self.k_params['loss']      + (1 -
beta) * kl

        self.k_params_prev = dict(self.k_params)
        self.k_params = {'bandwidth': kb, 'delay': kd,
'jitter': kj, 'loss': kl}

        if getattr(self, 'debug_mode', False):
            print(
                f"k-updated → kb={kb:.2f}, kd={kd:.2f},
kj={kj:.2f}, kl={kl:.2f} | "
                f"EMA: BW={self._ema_bw:.1f},
D={self._ema_d*1000:.1f}ms, "
                f"J={self._ema_j*1000:.1f}ms,
L={self._ema_l*100:.2f}%"
            )
```

```
        self._routing_dirty = True


    def finalize_pheromones(self):
        print(f"✅ Pheromones retained | EIGRP:
{self.eigrp_pheromone:.2f}, OSPF: {self.ospf_pheromone:.2f}")

    def make_protocol_decision(self):
        # تأكيد توافق البروتوكول مع نوع البيانات والهدف
        if self.data_type == DataType.VIDEO:
            if 'Video_Client' in self.destination or
'Mixed_Client' in self.destination:
                self.protocol_decision = Protocol.OSPF
            else:
                self.protocol_decision = Protocol.UNDEFINED

        elif self.data_type in [DataType.VOICE,
DataType.TEXT]:
            if ('Voice_Client' in self.destination or
'DB_Client' in self.destination or
                'Mixed_Client' in self.destination):
                self.protocol_decision = Protocol.EIGRP
            else:
                self.protocol_decision = Protocol.UNDEFINED
        else:
            self.protocol_decision = Protocol.UNDEFINED

        print(f"⚖️ Protocol → {self.protocol_decision.value} |
EIGRP: {self.eigrp_pheromone:.2f} | OSPF:
{self.ospf_pheromone:.2f}")
        return self.protocol_decision
```

```
    def calculate_quality_factor(self, conditions:
NetworkConditions, k: float = 1.0) -> float:
        import numpy as np

        # --- نوع الخدمة ---
        DataType_ = globals().get('DataType', None)
        svc = getattr(self, 'data_type', None)
        if svc is None and hasattr(self, 'ants') and
getattr(self, 'ants', None):
            svc = getattr(self.ants[0], 'data_type', None)
        if svc is None and DataType_ is not None:
            svc = DataType_.VIDEO

        def _upper(x):
            try:
                return str(x.name).upper()
            except:
                return str(x).upper()

        svc_name = _upper(svc)

        # --- التطبيع ITU ( [1..0] إلى") ---
        bw_cap = 4000.0 if svc_name == 'VIDEO' else (100.0 if
svc_name == 'VOICE' else 50.0)
        normalized = {
            'bandwidth': min(1.0, conditions.bandwidth /
bw_cap),
            'delay': 1.0 if conditions.delay * 1000 <= 150
else
                     0.0 if conditions.delay * 1000 >= 400
else
                     1.0 - ((conditions.delay * 1000 - 150) /
(400 - 150))**1.5,
```

```
            'jitter': 1.0 if conditions.jitter * 1000 <= 20
else
                      0.0 if conditions.jitter * 1000 >= 50
else
                      1.0 - ((conditions.jitter * 1000 - 20) /
(50 - 20))**1.3,
            'loss': 1.0 if conditions.loss * 100 <= 1.0 else
                    0.0 if conditions.loss * 100 >= 5.0 else
                    (1.0 - ((conditions.loss * 100 - 1.0) /
(3.0 - 1.0)) * 0.3) if conditions.loss * 100 <= 3.0 else
                    0.7 * (1.0 - (conditions.loss * 100 - 3.0)
/ (5.0 - 3.0))
        }

        # --- أوزان ITU ---
        if DataType_ is not None:
            weights = {
                DataType_.VIDEO: {'bandwidth': 0.6, 'delay':
0.2, 'jitter': 0.1, 'loss': 0.1},
                DataType_.VOICE: {'bandwidth': 0.1, 'delay':
0.3, 'jitter': 0.4, 'loss': 0.2},
                DataType_.TEXT:  {'bandwidth': 0.1, 'delay':
0.2, 'jitter': 0.0, 'loss': 0.7}
            }
            w = weights.get(getattr(DataType_, svc_name,
None),
                            {'bandwidth': 0.6, 'delay': 0.2,
'jitter': 0.1, 'loss': 0.1})
        else:
            w = {'bandwidth': 0.6, 'delay': 0.2, 'jitter':
0.1, 'loss': 0.1}

        a_b = float(getattr(self, 'a_b', 1.2))
```

```
        a_d = float(getattr(self, 'a_d', 1.2))
        a_j = float(getattr(self, 'a_j', 1.2))
        a_l = float(getattr(self, 'a_l', 1.2))

        if not hasattr(self, 'k_params') or not
isinstance(self.k_params, dict):
            self.k_params = {'bandwidth': 0.6, 'delay': 0.6,
'jitter': 0.6, 'loss': 0.6}
        k1 = float(np.clip(self.k_params.get('bandwidth',
0.6), 0.01, 5.0))
        k2 =
float(np.clip(self.k_params.get('delay',     0.6), 0.01, 5.0))
        k3 =
float(np.clip(self.k_params.get('jitter',    0.6), 0.01, 5.0))
        k4 =
float(np.clip(self.k_params.get('loss',      0.6), 0.01, 5.0))

        Ub, Ud, Uj, Ul = normalized['bandwidth'],
normalized['delay'], normalized['jitter'], normalized['loss']
        Ubt = 1.0 - np.exp(-k1 * a_b * Ub)
        Udt = 1.0 - np.exp(-k2 * a_d * Ud)
        Ujt = 1.0 - np.exp(-k3 * a_j * Uj)
        Ult = 1.0 - np.exp(-k4 * a_l * Ul)

        quality = (
            w['bandwidth'] * Ubt +
            w['delay']     * Udt +
            w['jitter']    * Ujt +
            w['loss']      * Ult
        )
        quality = float(np.clip(quality, 0.0, 1.0))

        # --- عامل العمر ---
```

```
        age_factor = max(0.3, 1.0 - (self.get_age() / 300))
        q_age = float(np.clip(quality * age_factor, 0.0, 1.0))

        alpha = float(getattr(self, 'ema_alpha', 0.3))
        if not hasattr(self, '_ema_Q'):
            self._ema_Q = q_age
        else:
            self._ema_Q = alpha * q_age + (1 - alpha) *
self._ema_Q

        if not hasattr(self, 'qos_samples'):
            self.qos_samples = []
        sample = {
            'bandwidth': conditions.bandwidth,
            'delay': conditions.delay,
            'jitter': conditions.jitter,
            'loss': conditions.loss
        }
        self.qos_samples.append(sample)
        try:
            NETWORK_SAMPLES.append(sample)
        except NameError:
            pass

        return self._ema_Q

    def get_age(self) -> float:
        return time.time() - self.creation_time

class Node:
    def __init__(self, node_id: str, node_type: NodeType):
        self.id = node_id
        self.type = node_type
        self.service_type = self._determine_service_type()
```

```
        self.network = None
        self.active_protocol = Protocol.UNDEFINED
        self.neighbors = []
        self.base_processing_delay = 0.05 if node_type ==
NodeType.ROUTER else 0.02
        self.current_load = 0.0
        self.conditions = NetworkConditions(
        delay=0.02 if node_type == NodeType.ROUTER else 0.01,
        bandwidth=2000 if node_type == NodeType.SERVER else
500,
        loss=0.005,
        jitter=2.0
    )


    def _determine_service_type(self) -> Optional[DataType]:
        if self.type != NodeType.CLIENT:
            return None
        if 'Video' in self.id:
            return DataType.VIDEO
        elif 'Voice' in self.id:
            return DataType.VOICE
        elif 'DB' in self.id:
            return DataType.TEXT
        return None

    def set_network(self, network: 'Network'):
        self.network = network

    def add_neighbor(self, neighbor_id: str):
        if neighbor_id not in self.neighbors:
            self.neighbors.append(neighbor_id)

    def get_processing_delay(self) -> float:
```

```
        return self.base_processing_delay * (1 +
self.current_load)

    def get_next_hop(self, destination: str, ant:
Optional[Ant] = None) -> Optional[str]:
        if not self.network:
            return None
        try:
            path = nx.shortest_path(self.network.graph,
self.id, destination, weight='delay')
            if len(path) > 1:
                next_hop_candidates = path[1:]
                for candidate in next_hop_candidates:
                    if candidate in self.neighbors:
                        return candidate
                return next_hop_candidates[0]
        except nx.NetworkXNoPath:
            pass

        if self.neighbors:
            return random.choice(self.neighbors)

        return None

    def receive_ant(self, ant: Ant):
        # التحقق من توافق الخدمة عند الوصول للهدف
        if self.id == ant.destination:
            is_video_server = 'Video' in ant.source
            is_video_client = 'Video' in self.id or 'Mixed' in
self.id

            # حالة 1: نملة من سيرفر فيديو إلى عميل غير فيديو
            if is_video_server and not is_video_client:
```

```
                print(f"❌ Ant {ant.id} invalid (Video server
to non-video client) - TERMINATED")
                ant.status = "FAILED"
                self.network.ant_completed(ant, False)
                return

            # حالة 2: نملة من سيرفر غير فيديو إلى عميل فيديو
            if (not is_video_server) and ('Video' in self.id):
                print(f"❌ Ant {ant.id} invalid (Non-video
server to video client) - TERMINATED")
                ant.status = "FAILED"
                self.network.ant_completed(ant, False)
                return

        processing_delay = self.get_processing_delay()
        ant.record_hop(self.id, self.conditions,
processing_delay)

        if ant.phase == AntPhase.EXPLORATION:
            self._handle_exploration_phase(ant)
        elif ant.phase == AntPhase.RETURN:
            self._handle_return_phase(ant)
        elif ant.phase == AntPhase.DECISION:
            self._handle_decision_phase(ant)

        self.current_load = max(0.0, self.current_load - 0.03)

    def _handle_exploration_phase(self, ant: Ant):
        if ant.current_node != self.id:
            ant.move_to_node(self.id)
        ant.update_pheromones(self.conditions)

        if self.id == ant.destination:
```

```
            if not ant.record_path(self.id):
                ant.status = "FAILED"
                self.network.ant_completed(ant, False)
                return
            ant.phase = AntPhase.RETURN
            ant.return_path = list(reversed(ant.current_path))
            ant.current_path.clear()
            self.forward_ant(ant)
            return

        if not ant.dfs_stack:
            nbrs = ant.get_unvisited_neighbors(self.network)
            if not nbrs:
                if len(ant.current_path) > 1:
                    last = ant.current_path.pop()
                    print(f"🔙 Backtracking from dead-end at
{last}")
                    ant.current_node = ant.current_path[-1]
                    self.network.nodes[ant.current_node].recei
ve_ant(ant)
                else:
                    ant.status = "FAILED"
                    self.network.ant_completed(ant, False)
                return
            ant.dfs_stack.append((self.id, nbrs))

        while ant.dfs_stack:
            node_id, nbrs = ant.dfs_stack[-1]
            if not nbrs:
                ant.dfs_stack.pop()
                if ant.current_path:
                    bk = ant.current_path.pop()
                    print(f"🔁 Backtracked from {bk} to
{ant.current_path[-1] if ant.current_path else 'START'}")
```

```
                    if ant.current_path:
                        prev = ant.current_path[-1]
                        ant.current_node = prev
                        self.network.nodes[prev].receive_ant(a
nt)
                else:
                    ant.status = "FAILED"
                    self.network.ant_completed(ant, False)
                return

            nxt = nbrs.pop(0)
            if ant.visited_times[nxt] < 2:
                ant.visited_times[nxt] += 1
                ant.current_node = nxt
                ant.current_path.append(nxt)
                ant.visited_nodes.add(nxt)
                next_nbrs =
ant.get_unvisited_neighbors(self.network)
                ant.dfs_stack.append((nxt, next_nbrs))
                self.network.nodes[nxt].receive_ant(ant)
                return


    def _handle_return_phase(self, ant: Ant):
        if ant.current_node == ant.source:
            print(f"🔁 Ant {ant.id} returned to source")
            ant.finalize_pheromones()
            ant.make_protocol_decision()
            ant.full_path =
list(dict.fromkeys(ant.forward_path + ant.return_path))
            ant.phase = AntPhase.DECISION
            ant.current_node = ant.destination
            self.forward_ant(ant)
        else:
```

```
            if ant.return_path:
                next_hop = ant.return_path.pop(0)
                ant.current_node = next_hop
                self.network.nodes[next_hop].receive_ant(ant)
            else:
                print(f"❌ Ant {ant.id} return_path exhausted
- dropped")
                ant.status = "FAILED"
                self.network.ant_completed(ant, False)

    def _handle_decision_phase(self, ant: Ant):
        if self.type == NodeType.ROUTER:
            self.active_protocol = ant.protocol_decision
            print(f"\033[1m🖧\033[0m Router {self.id} set to
{ant.protocol_decision.value}")

        if ant.full_path:
            current_index = ant.full_path.index(self.id)
            if current_index + 1 < len(ant.full_path):
                next_hop = ant.full_path[current_index + 1]
                ant.current_node = next_hop
                self.network.nodes[next_hop].receive_ant(ant)
                return

        ant.status = "COMPLETED"
        print(f"✅ Ant {ant.id} completed decision phase and
reached destination.")
        self.network.ant_completed(ant, success=True)

    def forward_ant(self, ant: Ant):
        try:
            # نحدد الهدف بناءً على مرحلة النملة (اكتشاف أو عودة)
```

```
            target = ant.destination if ant.phase !=
AntPhase.RETURN else ant.source
            # نحصل على العقدة التالية بناءً على الهدف
            next_hop = self.get_next_hop(target, ant)
            if next_hop and next_hop in self.network.nodes:
                ant.current_node = next_hop
                self.network.nodes[next_hop].receive_ant(ant)
                return True
            print(f"⚠ Ant {ant.id} failed to move from
{ant.current_node}")
            ant.status = "FAILED"
            self.network.ant_completed(ant, False)
            return False
        except Exception as e:
            print(f"⛔ Error forwarding ant {ant.id}:
{str(e)}")
            ant.status = "FAILED"
            self.network.ant_completed(ant, False)
            return False
```

Abstract

Computer communication networks constitute the backbone of modern information exchange infrastructure, offering a wide range of services including voice and video communication, file transfer, and interactive applications. To ensure efficient and reliable data transmission, these networks rely on various routing protocols such as OSPF, EIGRP, and RIP, which determine the paths that data packets follow from source to destination. Despite their effectiveness, these protocols are typically applied uniformly across the entire network without differentiating between data types or service-specific requirements, which may lead to suboptimal network performance.

In this study, we propose an enhanced hybrid routing algorithm inspired by the Ant Colony Optimization (ACO) technique. The algorithm aims to make routing decisions more intelligent and adaptive to the characteristics of the transmitted data. It identifies all possible complete paths between source and destination, and then applies the most suitable routing protocol—either OSPF or EIGRP—based on the nature of the data and the service requirements.

The core concept of the proposed algorithm lies in integrating the robustness of traditional routing protocols with the flexibility of artificial intelligence techniques. This fusion enables improved service efficiency across the network and delivers performance that is better aligned with the diverse demands of modern applications.

Keywords: Artificial Intelligence Algorithms, Interior Routing Protocols OSPF, EIGRP, RIP, Ant Colony Optimization (ACO)

Syrian Arab Republic

Damascus University

Faculty of Mechanical and Electrical Engineering

Computer and Automation Engineering Department

Developing algorithms of artificial intelligence to choose the appropriate routing protocol based on influencing factors

A dissertation submitted in fulfilment of the requirements for the degree of (PhD) in Computer Engineering and Networks

By

Eng. Mhd. Orfan Alkhatib

Supervisor

Dr. Bassam Mohamad

2024-2025