الجامعة الافتراضية السورية

SYRIAN VIRTUAL UNIVERSITY

الشبكات الحاسوبية

الدكتور غسان سابا

Books

الشبكات الحاسوبية

الدكتور غسان سابا

من منشورات الجامعة الافتراضية السورية

الجمهورية العربية السورية 2018

هذا الكتاب منشور تحت رخصة المشاع المبدع – النسب للمؤلف – حظر الاشتقاق (CC– BY– ND 4.0)

https://creativecommons.org/licenses/by-nd/4.0/legalcode.ar

يحق للمستخدم بموجب هذه الرخصة نسخ هذا الكتاب ومشاركته وإعادة نشره أو توزيعه بأية صيغة وبأية وسيلة للنشر ولأية غاية تجارية أو غير تجارية، وذلك شريطة عدم التعديل على الكتاب وعدم الاشتقاق منه وعلى أن ينسب للمؤلف الأصلي على الشكل الآتي حصراً:

غسان سابا، الشبكات الحاسوبية، من منشورات الجامعة الافتراضية السورية، الجمهورية العربية السورية، 2018

متوفر للتحميل من موسوعة الجامعة https://pedia.svuonline.org/

Computer Networks

Ghassan Saba

Publications of the Syrian Virtual University (SVU)

Syrian Arab Republic, 2018

Published under the license:

Creative Commons Attributions- NoDerivatives 4.0

International (CC-BY-ND 4.0)

https://creativecommons.org/licenses/by-nd/4.0/legalcode

Available for download at: https://pedia.svuonline.org/

الفهرس

الوحدة الأولى مقدمة عامة في الشبكات 1

مقدمة عن تراسل المعطيات DATA COMMUNICATION 2

فعالية انظمة تراسل المعطيات 2

مكونات أنظمة تراسل المعطيات 3

تدفق المعطيات 3

الشبكات NETWORKS 4

خواص الشبكة 4

نوع الاتصال 5

الطوبولوجية الفيزيائية Physical Topology 6

تصنيف الشبكات 8

الشبكات المحلية LAN 8

الشبكات الواسعة WAN 8

الشبكات الإقليمية MAN 9

الوحدة التانية النموذج المرجعي OSI 10

هيكلية الشبكات 11

النموذج المرجعي OSI 12

مفهوم العمليات من الند لند Peer-to-peer processes 13

الطبقة الفيزيائية Physical layer 13

طبقة وصلة المعطيات Data Link Layer 14

طبقة الشبكة Network Layer 15

طبقة النقل Transport Layer 16

طبقة الجلسات Session Layer 17

طبقة التقديم Presentation Layer 18

طبقة التطبيقات Application Layer 18

الوحدة الثالثة إكتشاف وتصحيح الخطاء ضمن طبقة وصلة المعطيات Data Link Layer 20

أنواع الأخطاء..... 21

أخطاء البث الواحد Single-Bit error..... 21

الخطأ الرشقي Burst Error 22

التكرار Redundancy 22

الرماز الكتلي Block Coding..... 23

إكتشاف الأخطاء..... 23

تصحيح الأخطاء..... 25

مسافة هامتك Hamming Distance..... 26

مسافة هامتك الصغرى..... 26

المسافة الصغرى لإكتشاف الأخطاء 27

الرماز الكتلي الخطي Linear Block Codes..... 27

بعض الرمازات الكتلية الخطية 28

رماز إختبار الزوجية المبسط Simple parity-check code..... 28

رمازات هامتك 30

الرمازات الدوراة Cyclic Codes..... 32

إختبار التكرارية الدوار(Cyclic Redundancy Check (CRC..... 32

المرمز Encoder 33

مفكك الترميز Decoder..... 34

تدقيق المجموع Checksum..... 35

الوحدة الرابعة التحكم ضمن طبقة وصلة المعطيات Data Link Control..... 37

التأطير Framing 38

الأطر ذات الطول الثابت Fixed-size framing 38

البروتوكولات محرفية التوجه Character-oriented protocols 38

طريقة حشو البايتات Byte-stuffing 39

البروتوكولات بتية التوجه Bit-oriented Protocols 40
طريقة الحشو البتي Bit-stuffing..... 40
بروتوكولات التحكم بالتدفق وبالأخطاء 41
التحكم بالتدفق..... 41
التحكم بالأخطاء 41
البروتوكولات..... 41
بروتوكولات القناة المثالية..... 41
البروتوكول المبسط 42
بروتوكول التوقف والانتظار Stop-and-wait..... 43
بروتوكولات القناة الغير المثالية..... 45
التوقف والانتظار مع طلب الإعادة الآلي Stop-and-Wait Automatic Repeat Request 45
الأرقام التسلسلية Sequence Numbers..... 45
أرقام الإقرارات Acknowledgment Numbers 46
فعالية البروتوكول..... 48
طلب الاعادة الآلي بالعودة ن – خطوة Go-Back-N ARQ 49
الأرقام التسلسلية Sequence Numbers..... 49
النافذة المنزلقة Sliding Window..... 49
المؤقت الزمني Timer 51
الاقرارات Acknowledgments..... 51
إعادة إرسال إطار..... 52
قياس نافذة الإرسال 52
بروتوكول تكرار الطلب الآلي مع تكرار انتقالي Selective Repeat Automatic Repeat Request 54
النافذة..... 54
تحليل البروتوكول..... 61
الوحدة الخامسة الشبكات المحلية LANs 64

معايير IEEE 65
التحكم بالوصلة المنطقية LLC..... 66
التحكم بالولوج الى الوسيط MAC 66
إيثرنت المعيارية 66
طبقة MAC الجزئية 67
بروتوكول CSMA/CD..... 67
العنونة Addressing..... 69
العناوين وحيدة الوجهة ومتعددة الوجهات والمعممة 70
الطبقة الفيزيائية..... 71
التغيرات التي طرأت على معايير إيثرنت 73
إستخدام الجسور 73
إيثرنت المبدلة Switched Ethernet..... 74
إيثرنت السريعة Fast Ethernet..... 75
طبقة MAC الجزئية 76
الطبقة الفيزيائية..... 76
الجيغابت إيثرنت Gigabit Ethernet..... 77
طبقة MAC الجزئية 77
عشرة جيغابت إيثرنت Ten-Gigabit Ethernet..... 78
الوحدة السادسة ترابط الشبكات Network Interconnections 80
أجهزة ترابط الشبكات Network Connection Devices 81
المجمعة الغير نشطة Passive hub..... 81
المكررات Repeaters 81
المجمعة النشطة Active Hub..... 82
الجسر Bridge..... 83
الجسور الثقافية Transparent Bridges..... 84

مشكلة الحلقة Loop 85
شجرة التغطية..... 87
إستخادم الجسور لربط شبكات محلية مختلفة..... 87
مبدلات الطبقة الثانية Layer-2 Switches 88
أجهزة ترابط الشبكات Network Interconnection Devices..... 90
المسيرات Routers..... 90
مبدلات الطبقة الثالثة Layer-3 Switches 90
العبارة Gateway..... 90
الشبكات المحلية الإفتراضية VLANS..... 90
الوحدة السابعة العنونة المنطقية لبروتوكول IPv4 92
عناوين IPv4 93
فضاء العنونة..... 93
التدوين Notation..... 93
العنونة الصفية Classful Addressing..... 94
معرف الشبكة ومعرف المضيف Netid and Hostid..... 95
مفهوم قناع الشبكة Net Mask 95
الشبكات الجزية Subnetting..... 96
العنونة اللاصفية Classless Addressing..... 96
كتل العناوين Address Blocks 96
قناع الشبكة Mask 97
الهرمية..... 98
هرمية متعددة المستويات 100
تخصيص العناوين 100
الوحدة الثامنة التوجيه والتيسيير Forwarding and Routing.....103
التسليم Delivery..... 104

التوجيه Forwarding 104
الطريقة الافتراضية Default Method 105
جداول التيسير Routing Tables 105
الجداول الساكنة 106
الجداول الديناميكية 106
صيغة جدول التيسيير 106
بروتوكولات التيسيير Routing Protocols 106
الإستمثال Optimization 107
التسيير المعتمد على شعاع المسافة Distance Vector Routing 107
الإقلاع Initialization 108
المشاركة Sharing 109
التحديث Updating 109
بروتوكول تسيير المعلومات RIP 111
التسيير المعتمد على حالة الوصلة Link State Routing 112
بناء جداول التسيير 113
البروتوكول المفتوح لإيجاد أقصر الطرق أولاً Open Shortest Path First (OSPF) 116

الوحدة الأولى

مقدمة عامة في الشبكات

١ مقدمة عن تراسل المعطيات Data Communication

تعريف : نقصد بتراسل المعطيات إمكانية إرسال أي نوع من المعلومات (نص، صوت، صورة، فيديو) بين جهازين باستخدام وسيط نقل ما كالكابلات.

حتى يتم تحقيق عملية التراسل يجب أن تكون جميع التجهيزات موجودة ضمن نظام اتصالات مكون من عتاديات Hardware وبرمجيات Software.

١ ١ فعالية أنظمة تراسل المعطيات

تتعلق فعالية نظام تراسل المعطيات بأربعة عوامل وهي:

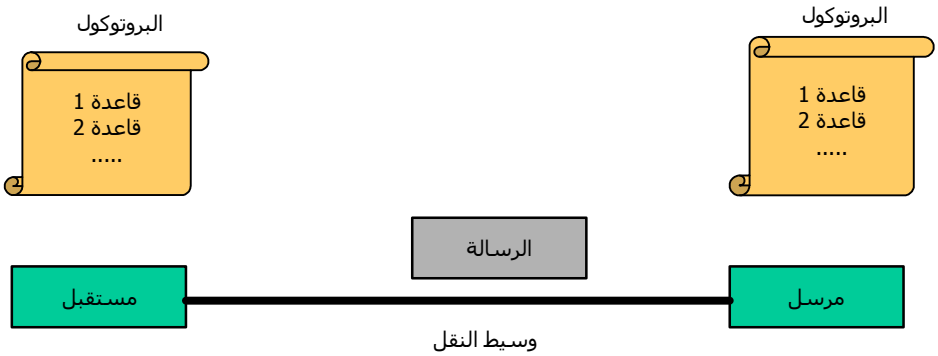
١. التوصيل Delivery: أي أنه يجب على النظام توصيل المعطيات إلى الوجهة الصحيحة دون غيرها.
٢. الدقة Accuracy: أي أنه يجب على النظام توصيل المعطيات كما أرسلت بدون أي تشويه أو تعديل.
٣. عدم التأخير : أي أنه يجب على النظام توصيل المعطيات بدون تأخير كبير وخاصةً في التطبيقات التفاعلية أو تطبيقات نقل الصوت والصورة.
٤. التذبذب Jitter: هو التغير الذي يطرأ على أزمنة وصول طرود المعلومات.

مثال: افترض أنك تريد إجراء مكالمة صوت وصورة مع صديق باستخدام الشبكة وأنك ترسل طرود المعلومات بمعدل طرد كل ٣٠ ميلي ثانية، فإذا تأخرت بعض الطرود ٤٠ ميلي ثانية فإن جودة العرض الحي سوف تتأثر سلباً بذلك.

تعريف طرد : عندما يريد طرف ما إرسال رسالة إلى طرف آخر فإنه يقوم بتقسيمها إلى أجزاء صغيرة وإرسال كل جزء ضمن طرد Packet الأمر الذي من شأنه تحسين أداء الشبكة.

١-٢ مكونات أنظمة تراسل المعطيات

تتألف أنظمة تراسل المعطيات من خمسة مكونات وهي:



الشكل ١- مكونات أنظمة تراسل المعطيات

١. الرسالة Message: تحوي المعلومات أو المعطيات المطلوب إرسالها. يمكن أن تحوي الرسالة على نص أو أرقام أو صور أو صوت أو مقاطع فيديو.
٢. المرسل Sender: هو الجهاز الذي يرسل المعطيات. يمكن أن يكون حاسوب أو كاميرة فيديو أو جهاز إنذار عن الحريق.
٣. المستقبل Receiver: هو الجهاز الذي يستقبل المعطيات. يمكن أن يكون حاسوب أو طابعة أو تلفزيون
٤. وسيط النقل Medium: هو المسار الفيزيائي الذي عن طريقه تستطيع الرسالة الانتقال من المرسل إلى المستقبل، طبعاً يمكن أن يكون وسيط النقل سلكياً مثل الكابلات النحاسية أو الكابلات المحورية أو الكابلات الضوئية أو لاسلكياً مثل الأمواج الراديوية.
٥. البروتوكول: هو مجموعة من القواعد التي تحكم إجراءات الاتصال بين جهازين.

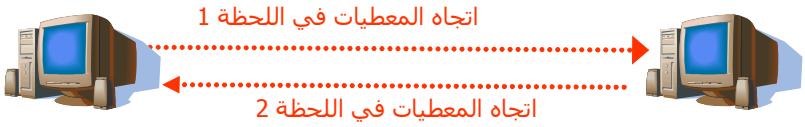
١-٣ تدفق المعطيات

توجد ثلاثة أنماط لتدفق المعطيات بين جهازين: البسيط ونصف المزدوج المتعاقب والمزدوج.

١. التراسل البسيط Simplex: يكون تدفق المعطيات وفق هذا النمط باتجاه واحد دوماً. أحد الأجهزة يكون مرسلاً دوماً والجهاز الآخر يكون مستقبلاً دوماً.

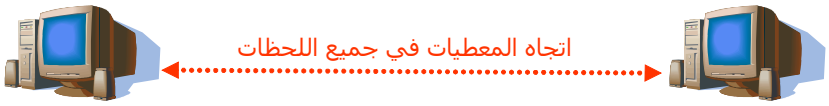
الشكل ٢- التراسل البسيط

٢. التراسل نصف المزدوج المتعاقب Half-duplex: يستطيع كل جهاز الإرسال والاستقبال لكن ليس في الوقت نفسه. أي عندما يرسل أحد الأجهزة فإن الجهاز الآخر يكون في حالة استقبال والعكس بالعكس. مثال: مرور سيارات على طريق بمسلك واحد مزود بإشارتين ضوئيتين، إشارة في كل طرف.



الشكل ٣– التراسل نصف المزدوج المتعاقب

٣. التراسل المزدوج Full-duplex: يستطيع كل جهاز الإرسال والاستقبال في نفس الوقت. مثال: طريق بمسلكين، ذهاب وإياب، يسمح بمرور السيارات في الاتجاهين بنفس الوقت. حتى نستطيع تحقيق هذا النوع من التراسل يجب إما تخصيص مسار فيزيائي مستقل لكل اتجاه أو تقسيم استطاعة القناة إلى جزأين حيث يمر كل جزء المعطيات باتجاه مختلف.



الشكل ٤– التراسل المزدوج

٢ الشبكات Networks

تعريف: الشبكة هي مجموعة من الأجهزة (أو العقد) المربوطة بوساطة وصلة اتصال. يمكن أن تكون العقدة حاسباً أو محطة عمل أو طابعة أو أي جهاز قادر على إرسال أو استقبال المعطيات.

٢–١ خواص الشبكة

يجب أن تتمتع الشبكة بالخواص التالية:

١. الأداء Performance: يمكن قياس الأداء بعدة طرق، نذكر منها زمن العبور Transit Time وزمن الاستجابة Response Time. زمن العبور هو الزمن الذي تستغرقه رسالة ما للعبور من جهاز إلى جهاز آخر، أما زمن الاستجابة فهو الزمن الفاصل بين إرسال طلب ما واستقبال الجواب عليه. يتعلق أداء الشبكة بمجموعة من العوامل مثل عدد المستثمرين ونوع وسيط النقل وإمكانيات العتاديات المربوطة إلى الشبكة وفعالية البرامج. هذا ويمكننا تقييم الأداء بشكل عام عن طريق النتاج Throughput وزمن التأخير Delay.

٢. الوثوقية reliability: تقاس وثوقية الشبكة بمدى تكرار حدوث الأعطال وبالزمن اللازم لتصليح العطل والعودة إلى الحالة الطبيعية ومدى متانة الشبكة وتحملها للكوارث الطبيعية.

٣. الأمن Security: نقصد بأمن الشبكات حماية المعطيات من الولوج غير المسموح به وحماية المعلومات من التخريب إضافةً إلى تحقيق سياسات أمنية وإجراءات لاستعادة الشبكة في حال حدوث اختراقات أمنية أو ضياع المعلومات.

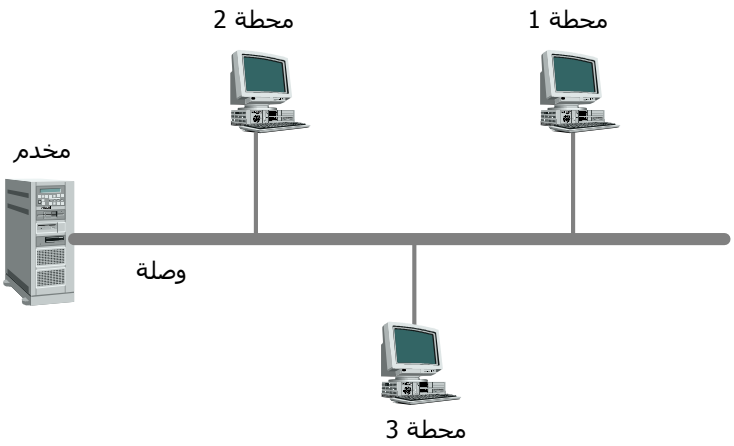
٢-٢ نوع الاتصال

يمكن أن تكون الشبكة مؤلفة من جهازين فقط أو مجموعة من الأجهزة. لذلك، فإما أن يكون الاتصال نقطة لنقطة Point to point أو متعدد النقاط Multi-point:

١. الاتصال نقطة لنقطة: يتم تحقيقه عن طريق توفير وصلة مخصصة بين الجهازين. تكون سعة الوصلة موضوعة تحت تصرف الجهازين بحيث لا تجري مشاركة هذه السعة مع أي جهاز آخر.
مثال: التحكم بالتلفزيون عن طريق جهاز التحكم عن بعد.

الشكل ٥- الاتصال نقطة لنقطة

٢. الاتصال متعدد النقاط: يتم تحقيقه عن طريق مشاركة أكثر من جهازين على وصلة واحدة. يجري في هذه الحالة تقاسم سعة الوصلة بين هذه الأجهزة.

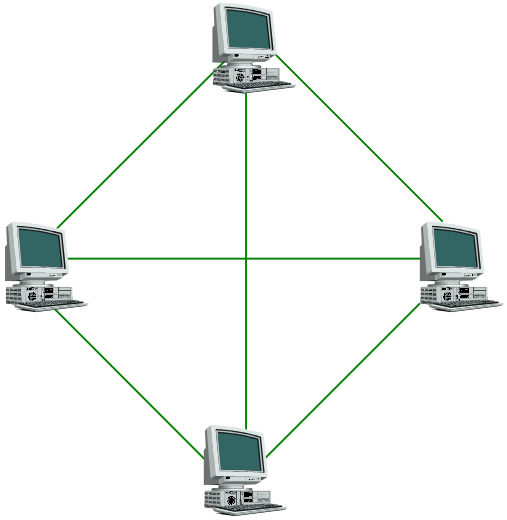


الشكل ٦- الاتصال متعدد النقاط

٢-٣ الطبولوجية الفيزيائية Physical Topology

نقصد بالطبولوجية الفيزيائية طريقة تمديد الشبكات فيزيائياً أي التمثيل الهندسي للعلاقة بين جميع الوصلات والأجهزة الموصولة ببعضها البعض. يوجد أربع طبولوجيات وهي العروية والنجمية والممرية والحلقية.

١. **العروية Mesh**: يجري هنا تخصيص وصلة نقطة لنقطة بين كل جهازين.

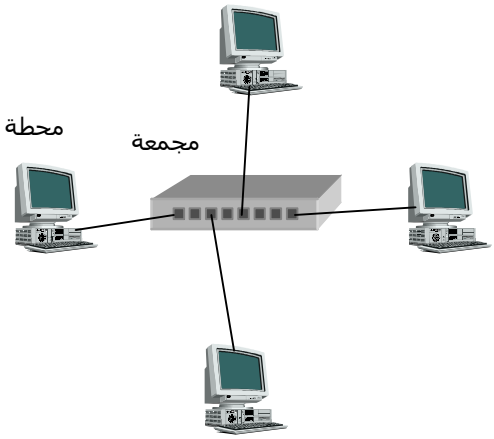


الشكل ٧- الطبولوجية العروية

أضف إلى ذلك أن كل جهاز يجب أن يمتلك (n-1) بوابة اتصال مع بقية الأجهزة.

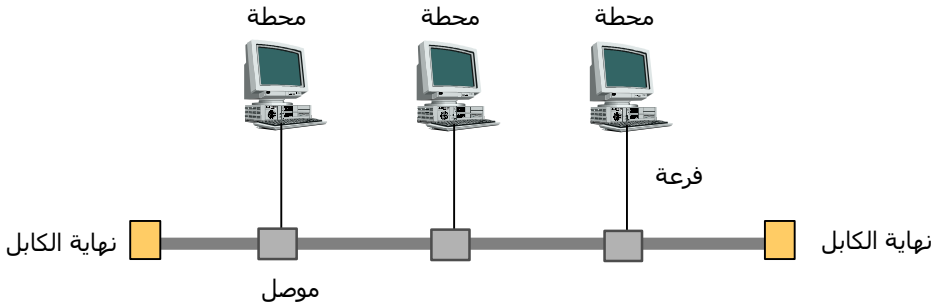
٢. **الطبولوجية النجمية Star**: يجري تخصيص كل جهاز بوصلة نقطة لنقطة إلى عقدة مركزية تدعى عادةً المجمعة Hub.

لاحظ أن الاتصال بين محطتين يجري عن طريق المجمعة وليس بشكل مباشر.



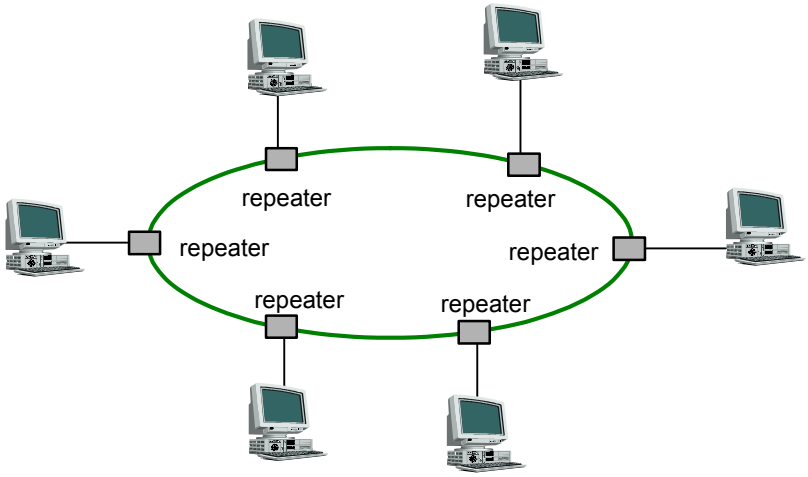
الشكل ٨- الطبولوجية النجمية

٣. **الطبولوجية الممرية Bus**: تعتمد الاتصال متعدد النقاط حيث يجري وصل كل محطة باستخدام فرعة Drop line وموصل Tap. تسمح الفرعة بوصل المحطة إلى الكابل الرئيسي بينما يسمح الموصل بإنشاء التماس الفيزيائي بين الناقلين الموجودين ضمن الفرعة والكابل.



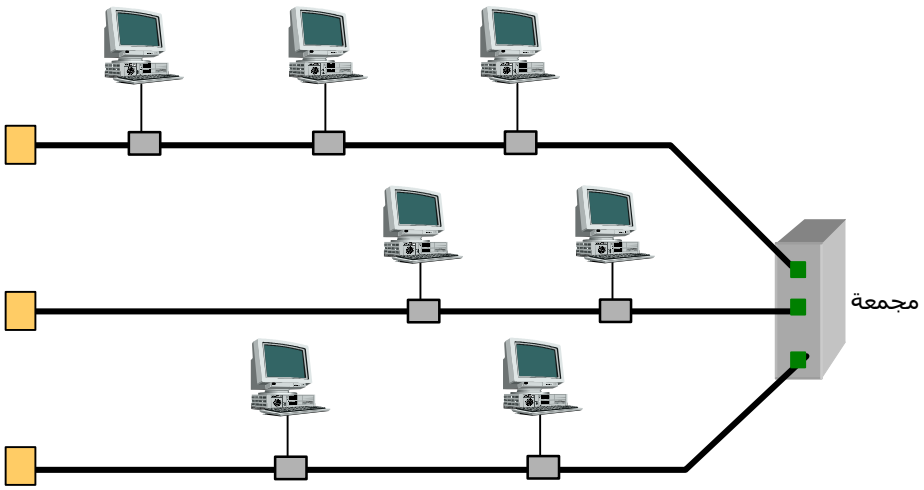
الشكل ٩– الطبولوجية الممرية

٤. **الطبولوجية الحلقية Ring**: يمتلك كل جهاز وصلتين نقطة لنقطة، الأولى مع الجهاز السابق والثانية مع الجهاز التالي. تنتقل الإشارات على الحلقة باتجاه واحد من جهاز إلى آخر حتى تصل إلى الوجهة النهائية. يكون مضمناً مع كل جهاز مكرر repeater يقوم، عند استقبال إشارة موجهة إلى جهاز آخر، بإعادة توليد الإشارة ومن ثم إرسالها.



الشكل ١٠– الطبولوجية الحلقية

٥. **الطبولوجية الهجينة Hybrid**: لا يوجد ما يمنع استخدام أكثر من طبولوجية واحدة لتحقيق شبكة.



الشكل ١١ – الطبولوجية الهجينة

٢-٤ تصنيف الشبكات

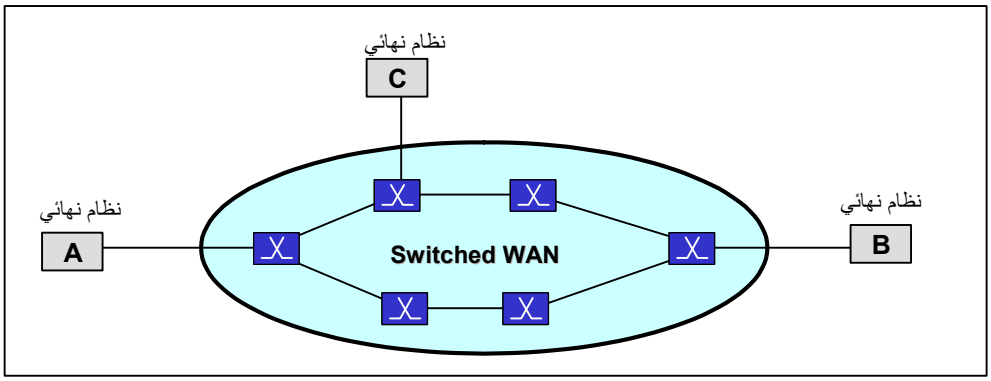
يتم تصنيف الشبكات حسب تغطيتها الجغرافية، فالشبكات ذات التغطية المحدودة والتي لا تتجاوز 3 km تدعى الشبكات المحلية Local Area Networks (LAN) والشبكات التي تغطي بلد كامل أو العالم ككل تدعى الشبكات الواسعة Wide Area Networks (WAN). كما يطلق على الشبكات ذات التغطية المتوسطة (مدينة على سبيل المثال) بالشبكات الإقليمية Metropolitan Area networks (MAN).

الشبكات المحلية LAN

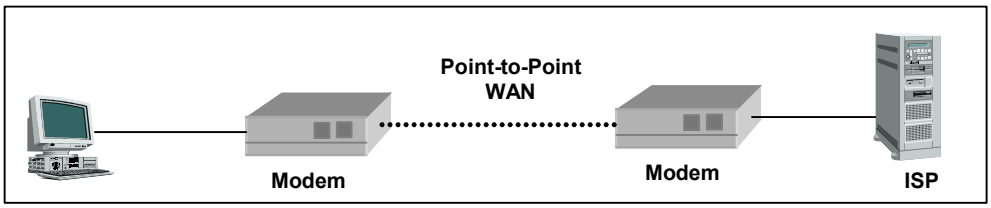
تكون الشبكات المحلية ذات ملكية خاصة حيث تربط التجهيزات ضمن مكتب واحد أو ضمن بناء أو ضمن حرم كامل. يجري تصميم الشبكات المحلية لتحقيق المشاركة على الموارد كالطابعات والبرامج أو على المعطيات. تتميز أخيراً الشبكات المحلية عن بقية أصناف الشبكات بوسيط النقل والطبولوجيات المستخدمة.

الشبكات الواسعة WAN

تزود الشبكات الواسعة إمكانية نقل المعلومات إلى مسافات بعيدة، يمكن أن تمتد على بلد أو على قارة أو حتى على العالم بأسره. يمكننا هنا التمييز بين الشبكة الواسعة المبدلة Switched WAN والشبكات الواسعة نقطة لنقطة كما هو موضح في الشكل التالي:



شبكة واسعة مبدلة



شبكة واسعة نقطة لنقطة

الشكل ١٢ – الشبكة الواسعة المبدلة والشبكة الواسعة نقطة لنقطة

لاحظ أنه يجري، في الشبكة الواسعة المبدلة (العليا)، وصل العقدة النهائية A إلى مبدلة تُربط بدورها إلى مبدلة أخـــرى وهكذا حتى نصل إلى العقدة النهائية في الطرف الآخر B أو أي عقدة نهائية أخرى مثل C. بينما يجري، في الـــشبكة الواسعة نقطة لنقطة وصل العقد النهائية باستخدام موديمات وشبكة واسعة نقطة لنقطة.

الشبكات الإقليمية MAN

يوجه هذا النوع من الشبكات إلى الزبائن الذين يحتاجون سرعة عالية لنقل المعطيات ويمتلكون عقد نهائية موزعة على مدينة.

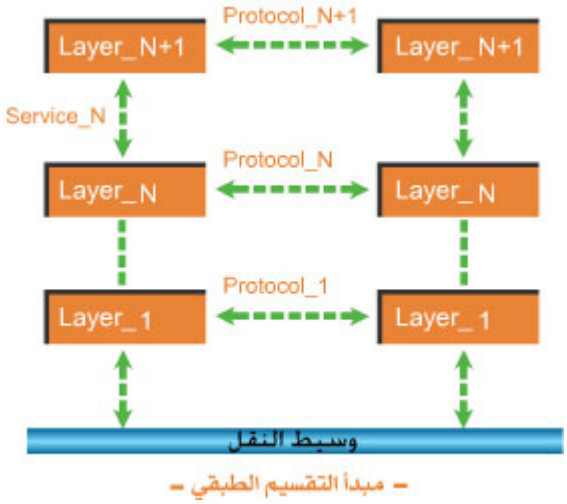
الوحدة الثانية

النموذج المرجعي OSI

1. هيكلية الشبكات

يجري، من أجل تبسيط التصميم، تقسيم الشبكات إلى طبقات Layers. يتعلق عدد هذه الطبقات وأسماؤها ووظائفها بنوع الشبكة.

تهدف كل طبقة، بشكل عام، إلى تزويد بعض الخدمات إلى الطبقة الأعلى منها مباشرة. كما أن كل طبقة (Layer_N) تدير عملية التخاطب مع طبقة من المستوى نفسه (الند) في محطة أخرى. تطلق عادة على قواعد واصطلاحات ومفردات التخاطب اسم بروتوكول المستوى N.



مثال: يبين الشكل التالي الطبقات الافتراضية المستخدمة لإرسال رسالة بين مرسل ومستقبل عن طريق البريد العادي.

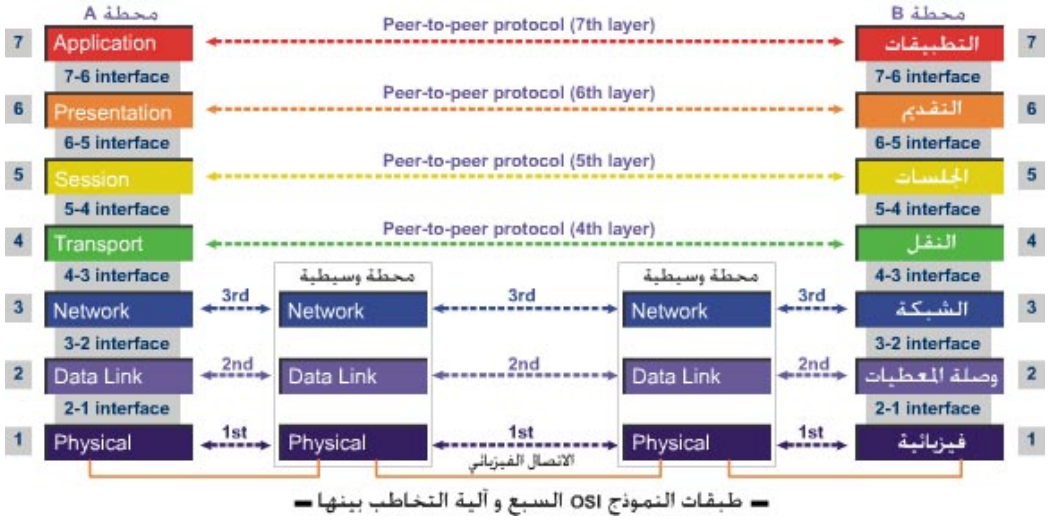
2. النموذج المرجعي OSI

عرفت منظمة المعايير العالمية International Standards Organization (ISO)، وهي منظمة دولية متخصصة بوضع المعايير العالمية التي تحظى بالقبول في شتى أنحاء العالم، نموذجاً لترابط الأنظمة المفتوحة Open Systems Interconnection (OSI) عام 1970. نقصد بالنظام المفتوح مجموعة البروتوكولات التي تسمح بتحقيق الاتصال بين أي نظامين مختلفين بغض النظر عن بنيتيهما الداخلية. ضمن هذا المنظور، يهدف نموذج OSI إلى تسهيل عملية الاتصال بين الأنظمة المختلفة دون الحاجة إلى إجراء تعديلات على مكوناتها العتادية أو البرمجية.

إن النموذج OSI ليس ببروتوكولاً وإنما نموذجاً لفهم وتصميم الشبكات تصميماً مرناً ومتيناً وقابلاً للتعامل مع شبكات أخرى.

يتألف النموذج OSI من سبع طبقات منفصلة ومتكاملة مع بعضها البعض، تسمح كل طبقة بتحقيق جزء من عملية نقل المعلومات عبر الشبكة.

يبين الشكل التالي طرق تحقيق ترابط الشبكات بين نظامين بعد المرور بمجموعة من المحطات الوسيطة.



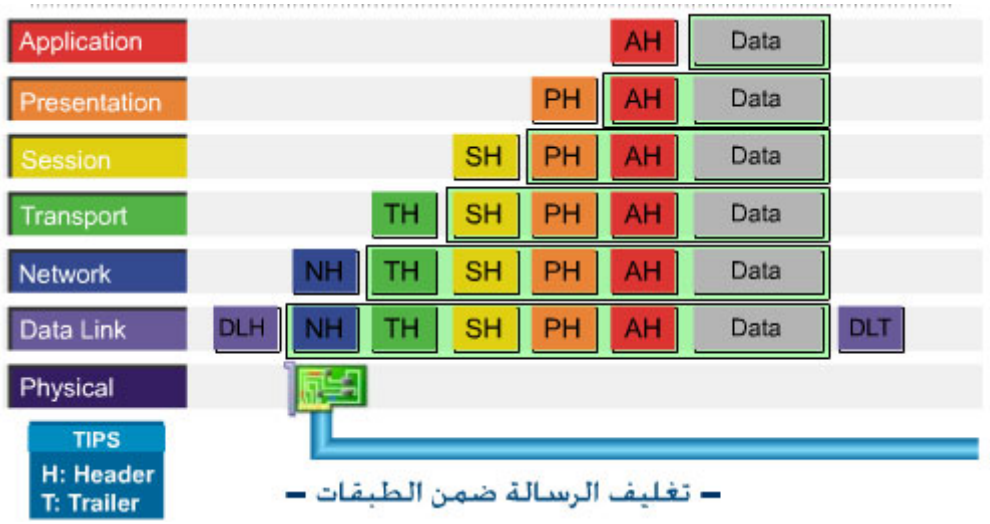
طبقات النموذج OSI السبع و آلية التخاطب بينها

جرى تحديد عدد الطبقات ووظيفة كل طبقة تحديداً يسمح بما يلي:

- يجب أن تحقق كل طبقة وظيفة معينة.
- يجب أن تُختار الحدود بين الطبقات بشكل يسمح بتقليل تدفق المعطيات بينها.
- يجب أن يكون عدد الطبقات كبير بشكل يسمح بتجنب وضع وظائف متغايرة في الطبقة نفسها.

مفهوم العمليات من ند لند Peer-to-peer processes

يجري الاتصال بشكل مباشر في الطبقة الفيزيائية، أي أن المحطة A ترسل سلسلة البتات على شكل إشارة ما إلى المحطة B (مروراً بمحطات وسيطة). أما فيما يتعلق بالطبقات العليا، فإن الاتصال يجب أن يمر تزولاً عبر الطبقات الأدنى للمحطة A حتى الوصول إلى المحطة B (انظر الشكل) حيث يمر الاتصال صعوداً من طبقة إلى الطبقة الأعلى منها. تضيف كل طبقة لدى الجهاز المرسل معلومات خاصة بها إلى الرسالة التي تستقبلها من الطبقة الأعلى وتمرر الرسالة الناتجة إلى الطبقة الأدنى.



يجري ضمن الطبقة 1 تحويل كل طرود المعلومات مع الإضافات إلى صيغة قابلة للنقل إلى المحطة المستقبلة. تقوم المحطة المستقبلة بفك تغليف الرسالة طبقة بطبقة، حيث تقوم كل إجرائية بمعالجة المعلومات التي تعنيها.

تجري عملية التخاطب بين طبقة وطبقة أدنى منها من خلال واجهة تخاطبية Interface معرفة بعناية وبشكل مستقل عن التحقيق Implementation وذلك بغية تمكين تغيير تحقيق الطبقة بدون تغيير الطبقات المجاورة.

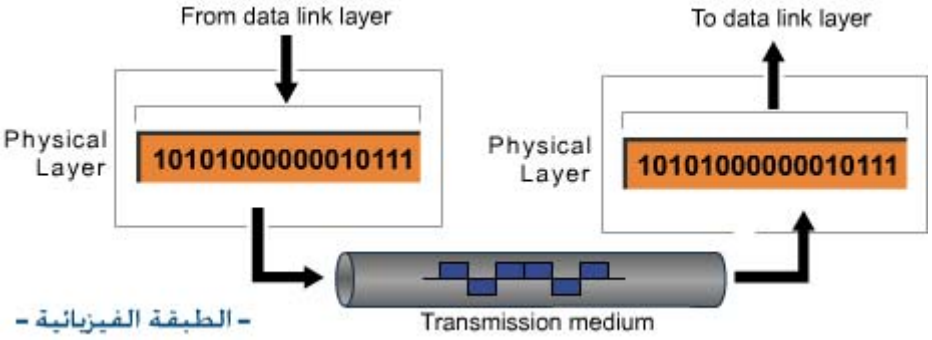
1. الطبقة الفيزيائية Physical layer

تهتم الطبقة الفيزيائية بنقل الخانات المنطقية bits على وسيط اتصال معين. لذلك فهي تُعنى بالمواصفات الكهربائية والإلكترونية والميكانيكية لوسطاء الاتصال communication media مثل:

- كم فولت لكل بت؟
- المدة الزمنية للبت.
- نوع الإرسال على الخط (مزدوج أو بسيط).

- نوع واجهة الاتصال مع الكابل Connectors interface وخصائصها الفيزيائية.
- تمثيل البتات وترميزها بإشارات كهربائية أو ضوئية حسب نوع وسيط النقل.
- مزامنة البتات: أي أنه يجب على المستقبل والمرسل، ليس فقط العمل على نفس السرعة، وإنما أن يكونوا متزامنين على مستوى البت أو على مستوى الساعة Clock
- إعداد الخط: هل الوصلة ستكون نقطة لنقطة أو متعددة النقاط؟
- تحديد الطبولوجية الفيزيائية للشبكة

لاحظ أن الطبقة الفيزيائية لا تضيف أي ترويسة للأطر القادمة من طبقة وصلة المعطيات.



2. طبقة وصلة المعطيات Data Link Layer

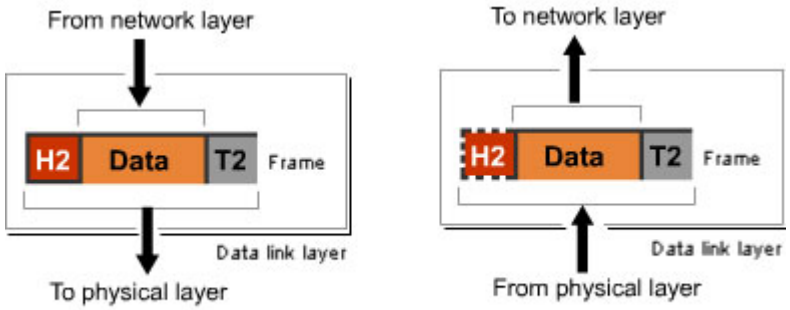
تهدف هذه الطبقة إلى حجب جميع مشاكل وأخطاء الطبقة الفيزيائية وتوفير خدمة نقل خالية من الأخطاء إلى طبقة الشبكة. لذلك تعمل على تأمين خدمة موثوقة نقطة لنقطة Point to point عن طريق تحقيق ما يلي :

- **التأطير Framing:** عن طريق تحويل البتات المستقبلة من الطبقة الفيزيائية إلى وحدات معطيات قابلة للإدارة تدعى بالأطر Frames.
- **العنونة الفيزيائية:** عندما تريد محطة ما إرسال إطار إلى وجهة تنتمي إلى نفس الشبكة فإن طبقة وصلة المعطيات تضيف ترويسة للتعريف عن المصدر والوجهة. أما إذا كانت الوجهة تنتمي إلى شبكة أخرى فيجري وضع العنوان الفيزيائي للمسير Router الافتراضي.
- **التحكم بالتدفق Flow Control:** إذا كانت سرعة استهلاك المعطيات لدى الوجهة أقل من سرعة إنتاج المعطيات لدى المصدر، فإن طبقة وصلة المعطيات تفرض استخدام آلية تحكم بالتدفق لتجنب إغراق الوجهة.

- **التحكم بالأخطاء Error Control:** تضيف طبقة وصلة المعطيات نوعاً من الوثوقية عن طريق إضافة آلية لاكتشاف ولإعادة إرسال الأطر المشوهة أو الضائعة. كما يمكنها أيضاً التعرف على الأطر المكررة.
- **التحكم بالولوج Access Control**: عندما يتم ربط جهازين أو أكثر إلى وسيط نقل مشترك فإنه يتوجب على طبقة وصلة المعطيات اختيار أي جهاز سيرسل في أي لحظة.

تجدر الإشارة هنا إلى أن طبقة وصلة المعطيات تضيف ترويسة ولاحقة للطرود القادمة من طبقة الشبكة.

يظهر الشكل التالي العلاقة بين طبقة وصلة المعطيات وطبقتي الشبكة والفيزيائية.

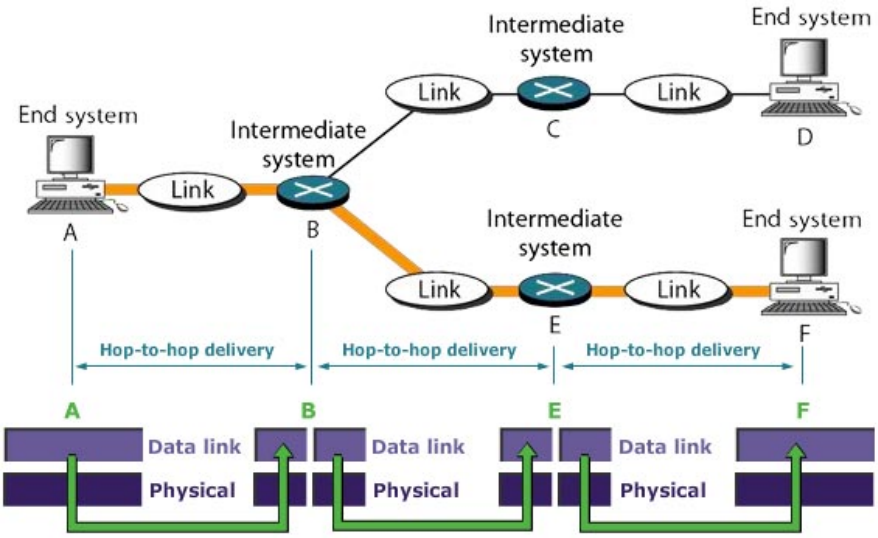


- طبقة وصلة المعطيات -

3. طبقة الشبكة Network Layer

تهتم هذه الطبقة بطريقة تسيير الطرود من المصدر إلى الوجهة بعد اجتياز مجموعة من الشبكات الوسيطة. لذلك تقوم بتحقيق ما يلي :

- **العنونة المنطقية Logical addressing:** يملك كل جهاز موصول إلى شبكة ما عنوانين: الأول فيزيائي محلي والثاني منطقي عام على جميع الشبكات.
- **التسيير Routing:** عندما يجري ربط أكثر من شبكة معاً لتشكيل ترابط شبكات Networks interconnection فإنه يتوجب على أجهزة الربط (المسيرات أو المبدلات) تسيير الطرود عبر الشبكات حتى تصل إلى الوجهة النهائية.
- **تحديد كون المسار معرف مسبقاً أو يحسب في حينه حسب ظروف الشبكة Static/Dynamic.**
- **تجزئة الرسائل الطويلة إلى طرود قصيرة وإعادة تجميعها لدى الوجهة.**
- **معالجة الاختناقات Congestion control التي يمكن أن تحدث في أوقات الذروة.**



طبقة الشبكة

يبين الشكل في الأعلى الحاجة إلى نوعين من الخدمات: الأول وهو قفزة لقفزة أو عقدة لعقدة Node to Node يؤمن الوثوقية بين كل عقدتين متتاليتين والثاني وهو نهاية لنهاية End to end يؤمن توصيل الطرد إلى وجهته النهائية.

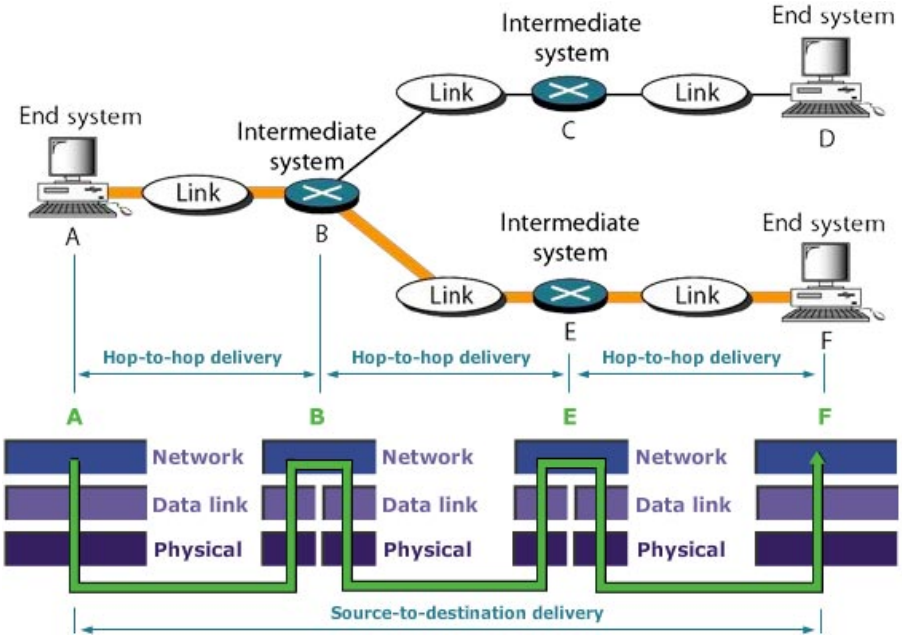
4. طبقة النقل Transport Layer

تؤمن هذه الطبقة نقل الرسالة بكاملها نقلاً موثوقاً وهي من نوع إجرائية لإجرائية Process to process. لاحظ أن الفرق بين طبقتي الشبكة والنقل يكمن في كون الأولى تعالج كل طرد على حدا وتنقله بشكل مستقل عن بقية الطرود دون معرفة العلاقة بين الطرود، بينما تضمن طبقة النقل وصول كامل الرسالة وصولاً سليماً ومرتباً بعد الإشراف على عمليات التحكم بالأخطاء والتدفق بين المصدر والوجهة.

المهام الأساسية لطبقة النقل:

- **عنونة نقاط الخدمة Service Point Addressing:** لا تقتصر عملية توصيل المعطات على الوصول إلى عنوان الوجهة فقط وإنما تعني أيضاً الوصول إلى الإجرائية (أي البرنامج التطبيقي) المطلوبة أيضاً. يجري تحديد عنوان الإجرائية المصدر أو الوجهة عن طريق معرفات Service Access Point (SAP) أو رقم البوابة Port Number.
- **التجزئة والتجميع Segmentation and Reassembly:** نقصد بذلك إمكانية أن تقوم طبقة النقل بتقسيم الرسالة المطلوب إرسالها إلى أجزاء قابلة للنقل وإضافة أرقام تسلسلية إلى هذه الأجزاء لتستطيع المحطة الوجهة إعادة تجميع الأجزاء واستخلاص الرسالة الأصلية.

- **التحكم بالوصلة Connection Control**: يمكن أن تكون طبقة النقل موصولة Connection-oriented أو غير موصولة Connection-less. عندما تكون طبقة النقل غير موصولة فإنها تعالج كل مقطع (يدعى الطرد على مستوى طبقة النقل بالمقطع Segment) من رسالة على أنه طرد مستقل وتوصله إلى طبقة النقل لدى الوجهة. بينما تؤسس طبقة النقل الموصولة اتصال مع طبقة النقل لدى الوجهة قبل البدء بتراسل الطرود وعند الانتهاء يجري قطع الاتصال.
- **التحكم بالتدفق:** كما هو عليه الحال ضمن طبقة وصلة المعطيات فإن طبقة النقل تقوم بالتحكم بالتدفق لكن من نهاية لنهاية وليس من نقطة لنقطة.
- **التحكم بالأخطاء:** كذلك الأمر بالنسبة للتحكم بالأخطاء الذي يجري أيضاً من نهاية لنهاية.

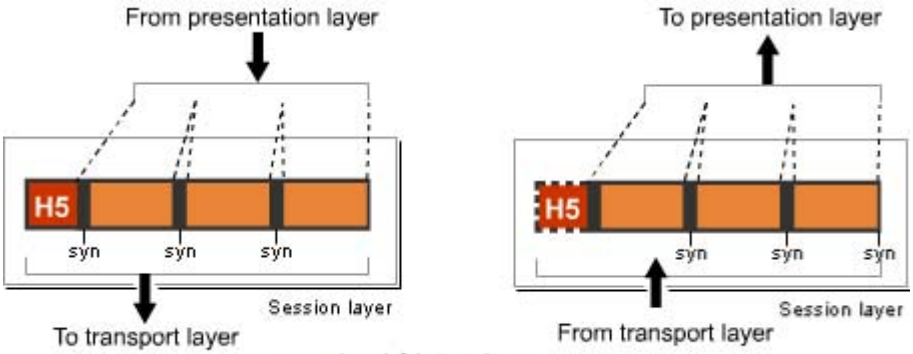


التسليم من المصدر إلى الوجهة

5- طبقة الجلسات Session Layer

تؤمن هذه الطبقة خدمات متقدمة لبعض التطبيقات، مثل:

- **إدارة الحوار:** تسمح طبقة الجلسات لنظامين بالدخول ضمن حوار. فهي تسمح بتحقيق التواصل بين إجرائيتين بشكل بسيط أو مزدوج.
- **التزامن وتحديد نقاط العودة:** تسمح طبقة الجلسة لإجرائية ما بإضافة نقاط تزامن إلى سلسلة معطيات؛ فإذا وقع عطل ما أثناء الإرسال يمكن العودة إلى آخر نقطة تزامن بدلاً من إعادة الإرسال من جديد.



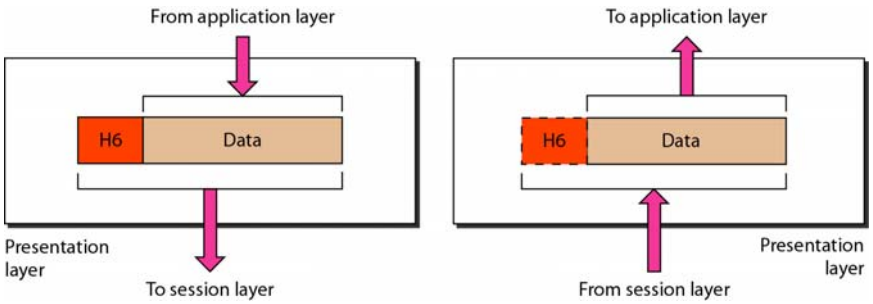
- طبقة الجلسات -

يبين الشكل العلوي ترويسة طبقة الجلسات HS ونقاط المزامنة (أو العودة) التي قامت بإضافتها.

6- طبقة التقديم Presentation Layer

تهتم هذه الطبقة بنحو Syntax ودلالة Semantic المعلومات المتراسلة بين نظامين. نذكر من المهام الأساسية لطبقة التقديم:

- الترجمة Translation: تسمح طبقة التقديم بتحقيق الترجمة بين ترميزات مختلفة للمعلومات.
- التعمية Encryption
- الضغط



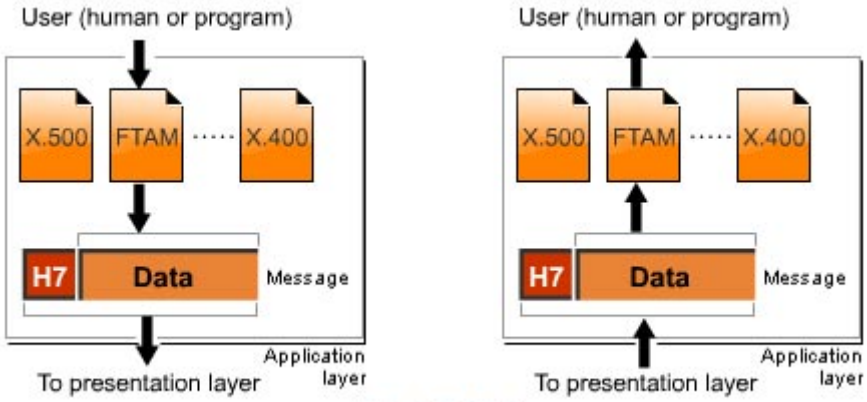
طبقة التقديم

7- طبقة التطبيقات Application Layer

تدير هذه الطبقة برمجيات المستثمر. من التطبيقات المستخدمة في هذه الطبقة :

- البريد الإلكتروني
- نقل الملفات والولوج إليها وإدارتها
- برمجيات البحث واستخلاص المعلومات

يبين الشكل في الأسفل طريقة التخاطب بين المستثمر النهائي (برنامج أو شخص) مع طبقة التطبيقات. لاحظ أنه النموذج المرجعي يستخدم المعيار X.400 للبريد الإلكتروني والمعيار File Transfer Access and Management (FTAM) لنقل الملفات وإدارتها ويستخدم معيار X.500 كخدمة الدليل Directory Services.



ـ طبقة التطبيقات ـ

الوحدة الثالثة

اكتشاف وتصحيح الأخطاء

ضمن طبقة وصلة المعطيات Data Link Layer

تهدف طبقة وصلة المعطيات إلى تحويل الطبقة الفيزيائية إلى وصلة مسؤولة عن تحقيق الاتصال من عقدة إلى عقدة Node to node. لذلك فإنها تهتم بالتأطير والعنونة الفيزيائية والتحكم بالتدفق والتحكم بالأخطاء والتحكم بالولوج إلى وسيط النقل. تقوم طبقة وصلة المعطيات عادةً بتجزئة الطرود القادمة من طبقة الشبكة (الأعلى) وتحويلها إلى وحدات معطيات يطلق عليه اسم "أطر Frames" تسهل إدارتها.

في كل مرة يجري فيها نقل المعطيات من جهاز إلى آخر فإنه يمكن أن تقع أخطاء في النقل أو تتشوه الطرود المنقولة نتيجةً لضجيج خارجي أو تضيع بعض الأطر. يجب أن تضمن الشبكات مطابقة المعلومات المستقبلة لما تم إرساله فعلاً. لذلك فإن غالبية التطبيقات تتطلب آليات لاكتشاف الأخطاء أو تصحيحها.

1. أنواع الأخطاء

تكون البتات، عندما تنتقل من نقطة إلى أخرى، عرضةً للتداخل interference. هذا التداخل يمكن أن يغير قيمة بت واحد أو مجموعة بتات.

1.1 أخطاء البت الواحد Single-Bit error

يجري هنا تغيير بت واحد فقط من سلسلة البتات المرسلة (بايت أو محرف أو أي وحدة معطيات أخرى) من الصفر المنطقي إلى الواحد أو بالعكس.يبين الشكل التالي مثالاً عن تحريف قيمة بتاً واحداً.

مثال 1:

بافتراض أن سرعة نقل المعطيات هي 1 Mbps وأن مصدر ضجيج خارجي يستمر لفترة 100 µs.

1. احسب عدد البتات التي يمكن أن تتشوه أثناء النقل
2. احسب فترة الضجيج اللازمة لوقوع خطأ ببت واحد فقط
3. استنتج مما سبق إمكانية حدوث الخطأ ببت واحد.

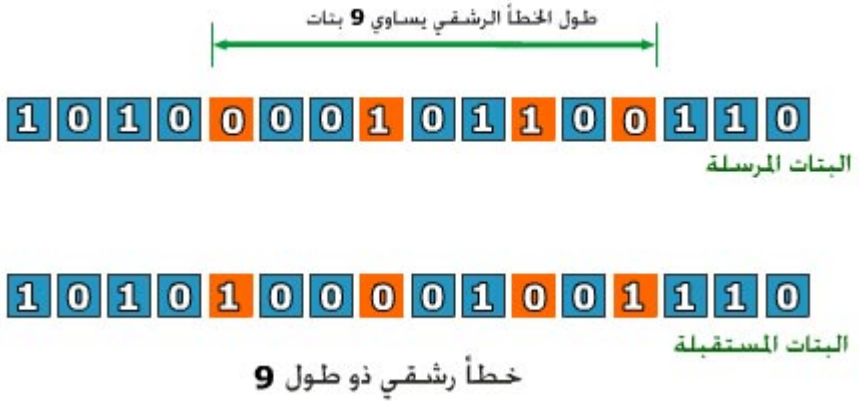
الحل:

بما أن معدل نقل المعطيات هو 1 Mbps فهذا يعني أن كل بت يستمر لفترة 1 µs. وبما أن الضجيج الخارجي يستمر لفترة 100 µs، ينتج من ذلك:

- عدد البتات التي يمكن أن تتشوه أثناء النقل = 100 بت
- فترة الضجيج اللازمة لوقوع خطأ على بت واحد هي 1 µs
- إمكانية وقوع خطأ على بت واحدة نادرة جداً (باحتمال 1%)

2.1 الخطأ الرشقي Burst Error

يعني تغيير قيمة بتين أو أكثر ضمن وحدة المعطيات المرسلة.



ملاحظة:

لا يعني الخطأ الرشقي تغيير قيم جميع البتات المتتابعة وإنما يقاس طول الخطأ بالمسافة الفاصلة بين أول بت وآخر بت خضعا للتغيير.

هذا النوع من الأخطاء هو الأكثر شيوعاً وذلك لأن الضجيج الخارجي يمكن أن يستمر لفترة زمنية أكبر من زمن إرسال بت واحد.

2. التكرار Redundancy

حتى نستطيع اكتشاف أو تصحيح الأخطاء فإننا نحتاج إلى استخدام بتات إضافية (تكرارية) مع المعطيات.

تعتبر مسألة تصحيح الخطأ أكثر تعقيداً من اكتشافه فقط، وذلك لأننا نحتاج لاكتشاف الخطأ جواباً من نوع نعم أو لا فقط حتى أننا لا نحتاج للتمييز بين خطأ واحد أو رشقي، بينما نحتاج لتصحيح الخطأ إضافةً إلى ما سبق تحديد المكان أو الأمكنة التي وقعت فيه الأخطاء لتصحيحها.

تمرين:

- عدد احتمالات تصحيح خطأ على بت واحد ضمن وحدة معطيات مؤلفة من 8 بتات.
- عدد احتمالات تصحيح خطأ على بتين ضمن وحدة معطيات مؤلفة من 8 بتات.

يوجد فعلياً طريقتين لتصحيح الأخطاء: الأول تدعى تصحيح الأخطاء المباشـر Forward Error Correction حيث يسعى المستقبل إلى معرفة مكان أو أمكنة الأخطاء وتصحيحها بينما يتم تحقيق النوع الثاني عن طريق إعادة الإرسال، أي اكتشاف الخطأ من قبل المستقبل وطلب إعادة الإرسال.

3. الرماز الكتلي Block Coding

في الرماز الكتلي، نقوم بتقسيم الرسالة الأصلية إلى كتل يتألف كل منها من k bits وتدعى كلمات المعطيات Datawords. نضيف بتات تكرارية r bits إلى كل كتلة بحيث يصبح المجموع n=k + r. تدعى الكتل الناتجة كلمات الرماز Codewords.

لاحظ هنا أنه يجري ترميز 2^k datawords باستخدام 2^k codewords صالحين ويبقى لدينا $2^n - 2^k$ Codewords غير صالحين.

مثال1:

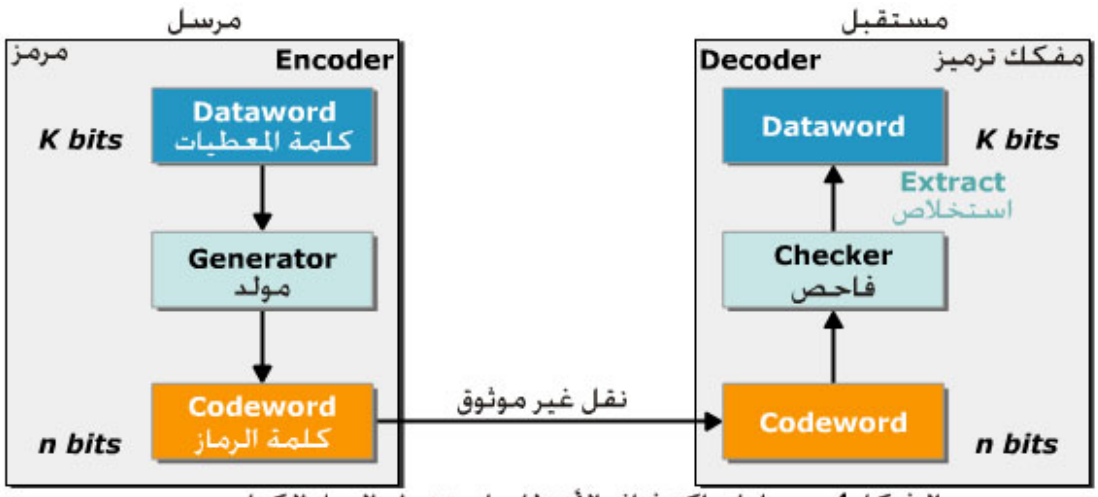
لنفترض أن k = 4 and n = 5 هذا يعني أنه يمكننا استخدام $2^4 = 16$ datawords وتعريف $2^5 =$ 32 codewords. أي أنه من بين 32 كلمة رماز فإننا نستخدم 16 لنقل الرسائل ونترك 16 كلمة رماز لأغراض أخرى كالتحكم واكتشاف الأخطاء.

1.3 اكتشاف الأخطاء

يمكن للمستقبل أن يكتشف حدوث تغيير في كلمة المعطيات إذا تحقق الشرطين التاليين:

- يمتلك المستقبل (أو يمكن أن يحصل على) لائحة بكلمات المعطيات الصحيحة.
- جرى تغيير كلمة الرماز الأصلية إلى كلمة رماز غير صالحة.

يبين الشكل التالي طريقة ترميز المعطيات وتفكيك ترميزها.



الشكل 4 - عمليات اكتشاف الأخطاء باستخدام الرماز الكتلي

لاحظ من الشكل العلوي أن المرسل يقوم باستخدام المولد Generator لحساب Codeword موافقة لـ dataword المطلوب إرسالها ومن ثم باستخدام وسيط النقل غير الموثوق لنقل كلمة الرماز إلى المستقبل. أما المستقبل فيقوم أولاً باختبار كون كلمة الرماز التي وصلت إليه صالحة أو لا؛ فإذا كانت صالحة فيجري قبولها واستخلاص كلمة المعطيات الموافقة لها أما إذا كانت غير صالحة فيتم إهمالها. في حال تعرضت كلمة الرماز المستقبلة لتشوه أثناء الإرسال وتغيرت إلى كلمة رماز صالحة أيضاً فلا يمكن للمستقبل اكتشاف هذا النوع من الأخطاء.

مثال2:

لنفترض أن $k = 2$ and $n = 3$. يبين الجدول التالي قائمة بالتقابلات بين كلمات المعطيات وكلمات الرماز.

Datawords	Codewords
0 0	0 0 **0**
0 1	0 1 **1**
1 0	1 0 **1**
1 1	1 1 **0**

الجدول 1 – رماز من أجل اكتشاف الأخطاء

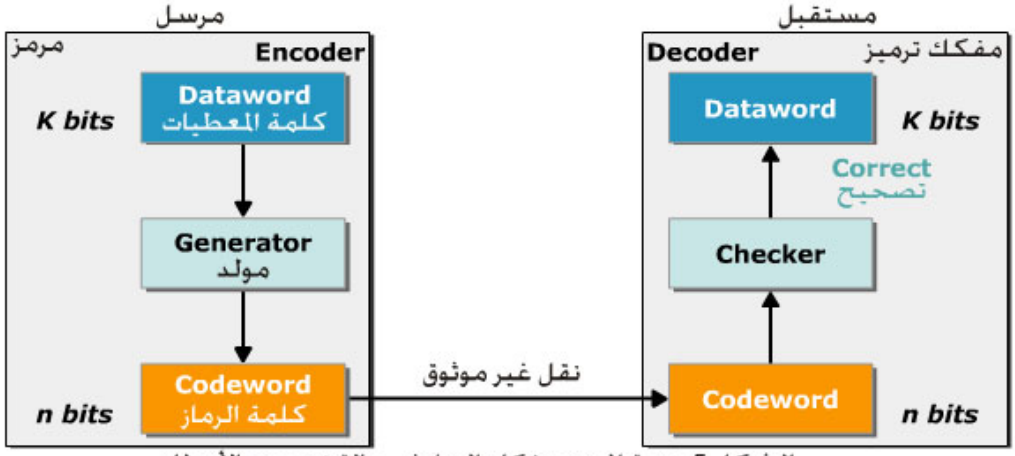
لنعتبر الحالات التالية:

- يستقبل المستقبل 011 التي هي صالحة. يستخلص منها المستقبل كلمة المعطيات 01
- تم تحريف كلمة الرماز أثناء الإرسال حيث تم استقبال 111 (أي جرى قلب قيمة آخر بت). إن كلمة الرماز هذه غير صالحة وسيتم إهمالها.

- يتم تحريف كلمة الرماز أثناء الإرسال لتصبح 000 (قلب قيمة بتين معاً). بما أن كلمة الرماز المستقبلة صالحة فسيتم قبولها بشكل خطئ من قبل المستقبل وسيستخلص منها القيمة 00. أي أن تحريفين متتاليين جعلا الخطأ غير قابل للاكتشاف.

2.3 تصحيح الأخطاء

إن مسألة تصحيح الخطأ تقتضي اكتشاف حالة الخطأ أولاً ومن ثم محاولة التعرف على البتات التي وقع عليها الخطأ. لذلك فإننا نحتاج على بتات تكرارية أكثر مما هو عليه الحال في اكتشاف الأخطاء فقط.



الشكل 5 - بنية المرمز ومفكك الرماز في حالة تصحيح الأخطاء

مثال3:

لنحاول إضافة بتات أخرى على المثال 3 لنرى هل يصبح ممكناً تصحيح الأخطاء. سنضيف 3 بتات تكرارية على البتين المكونين لكلمة المعطيات لتصبح كلمة الرماز مكونة من 5 بتات كما هو موضح بالجدول التالي:

Datawords	Codewords
0 0	0 0 0 0 0
0 1	0 1 0 1 1
1 0	1 0 1 0 1
1 1	1 1 1 1 0

الجدول 2 – رماز من أجل تصحيح الأخطاء

لنفترض أن المرسل يريد إرسال Dataword = 01 فهو يرمزها بـ Codeword = 01011 وإذا افترضنا أيضاً أن الرسالة قد حرفت أثناء النقل ووصل المستقبل الرسالة Codeword = 01001. يلاحظ المستقبل أولاً أن الكلمة الرمازية غير موجودة ضمن جدول التقابلات مما يعني أن خطأ

ما قد حدث. يقوم المستقبل عندئذ، مفترضاً أن بتاً واحداً فقط قد حرف، باستخدام الطريقة التالية ليخمن مكان الخطأ:

1. مقارنة الكلمة الرمازية المستقبلة مع أول كلمة رمازية ضمن القائمة (00000 مع 01001)، فيلاحظ أنها مختلفة عنها تماماً ولا يمكن أن تكون هي الكلمة المطلوبة
2. بالطريقة نفسها يستنتج المستقبل أن الكلمة الرمازية لا يمكن أن تكون هي الثالثة أو الرابعة ضمن القائمة
3. فإذاً يجب أن تكون الكلمة الرمازية هي الثانية لأنها الأقرب (تختلف عنها ببت واحد فقط). يقوم المستقبل بتغيير الكلمة الرمازية المستقبلة 01001 بالكلمة 01011 ويعود إلى القائمة لاستخلاص كلمة المعطيات وهي 01.

3.3 مسافة هامنك Hamming Distance

تعريف: مسافة هامنك بين كلمتين بطول متساوٍ هي عدد البتات المختلفة بين الكلمتين. وتكتب على الشكل التالي d(x, y) حيث x هي الكلمة الأولى وy هي الكلمة الثانية.

يمكننا حساب مسافة هامنك بين كلمتين بتطبيق (⊕) XOR على الكلمتين وحساب عدد الواحدات من النتيجة.

تدريب 1

1. احسب مسافة هامنك بين الكلمتين 011 and 000.
2. احسب مسافة هامنك بين الكلمات التالية:

1010 and 0100
11000 and 00101

4.3 مسافة هامنك الصغرى

يمكننا الآن تعريف مسافة هامنك الصغرى بين عدة كلمات رماز على أنها أصغر مسافة هامنك بين جميع الأزواج الممكنة للكلمات. سنستخدم الاصطلاح d_{min}.

تدريب

1. احسب مسافة هامنك الصغرى لقائمة الرماز المعرفة بالجدول 1.
2. أوجد مسافة هامنك الصغرى للجدول 2.

تذكر جيداً أن أي مخطط رماز يجب أن يحوي على ثلاثة وسطاء وهم

1. طول كلمة المعطيات **k**
2. طول كلمة الرماز **n**
3. مسافة هامنك الصغرى d_{min}.

يمكننا كتابة مخطط الرماز على الشكل **C(n, k)** مع عبارة مستقلة لـ d_{min}.

5.3 المسافة الصغرى لاكتشاف الأخطاء

حتى نستطيع اكتشاف خطأ يشمل s bits فإن مسافة هامنك الصغرى للرماز الصالح يجب أن تكون s + 1.

مثال5:

إن مسافة هامنك الصغرى في المثال 3 هي 2، هذا يعني أن مخطط الرماز المستخدم يضمن اكتشاف خطأ واحد فقط.

تدريب

1. أوجد عدد الأخطاء الذي يضمن اكتشافها مخطط الرماز في التمرين 3-2.

مثال6:

إذا كان لدينا مخطط رمازي ذو مسافة هامنك صغرى تبلـغ $d_{min} = 4$، أوجد عدد الأخطاء القادرعلى اكتشافها وتلك القادر على تصحيحها.

الحل

يضمن هذا المخطط اكتشاف حتى 3 أخطاء s = 3 لكنه يضمن تصحيح خطأ واحد فقط.

4. الرماز الكتلي الخطي Linear Block Codes

ضمن الرماز الكتلي الخطي، تطبيق XOR (addition modulo-2) على كلمتي رماز صالحتين يعطي كلمة رماز صالحة.

مثال7:

لنرى إذا كانت مخططات الرماز المستخدمة في الجداول 1 و2 هي كتلية خطية أو لا.

1. المخطط ضمن الجدول 1 هو كتلي خطي لأن نتيجة XOR لأي كلمتي رماز منه ينتج عنه كلمة رماز صالحة.

2. المخطط ضمن الجدول 2 هو أيضاً كتلي خطي.

المسافة الصغرى للرماز الكتلي الخطي: هو أقل عدد للواحدات الموجودة ضمن كلمة رماز لا تساوي الصفر.

مثال8:

المسافة الصغرى ضمن مخطط رماز الجدول 1 هي 2 والمسافة الصغرى ضمن مخطط رماز الجدول 2 هي 3.

1.4 بعض الرمازات الكتلية الخطية:

1.1.4 رماز اختبار الزوجية المبسط Simple parity-check code

يتم هنا استخدام كلمات معطيات ذات طول k وكلمات رماز ذات طول $n = k + 1$. يتم اختيار البت الإضافي بشكل يصبح عدد الواحدات الكلي ضمن كلمة الرماز زوجي.

تكون مسافة هامنك الصغرى لهذا النوع من الرماز $d_{min} = 2$، أي أنه يسمح باكتشاف خطأ على بت واحد ولا يسمح بتصحيح أي خطأ.

إن الرماز المستخدم ضمن الجدول 2 هو من هذا النوع.

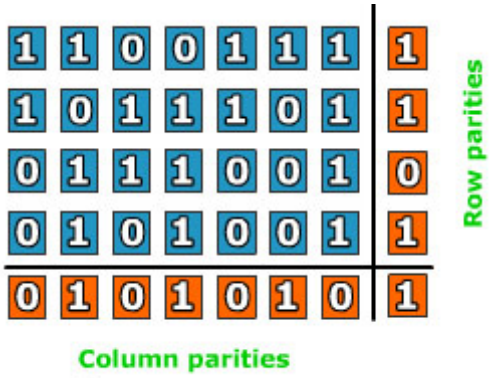
يسمح رماز اختبار الزوجية المبسط باكتشاف الأخطاء على الأعداد الفردية من البتات.

تدريب

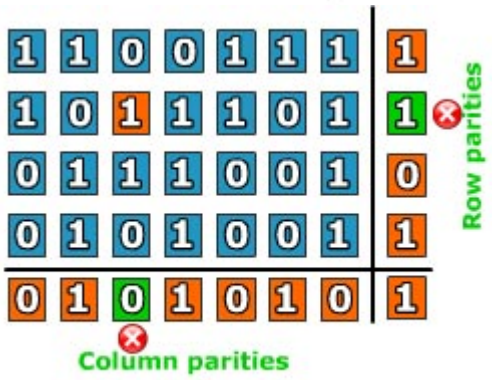
بافتراض أن المرسل ارسل كلمة المعطيات 1011، أوجد ما يلي:

1. كلمة الرماز المرسلة
2. طريقة المعالجة لدى المستقبل في حالة عدم تحريف المعطيات
3. طريقة المعالجة لدى المستقبل في حالة تحريف البت الثالث من كلمة الرماز
4. طريقة المعالجة لدى المستقبل في حالة تحريف البت الإضافي
5. طريقة المعالجة لدى المستقبل في حالة تحريف البت الثالث والبت الإضافي معاً
6. طريقة المعالجة لدى المستقبل في حالة تحريف البتات الأول والثاني والثالث.

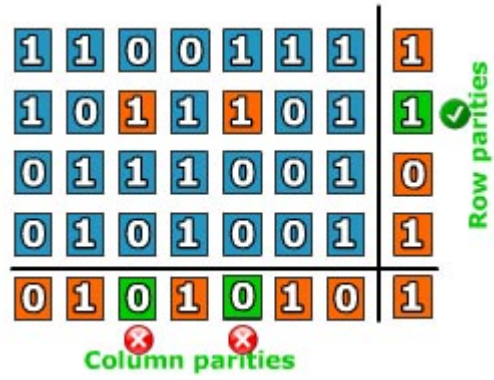
لاحظ أن النتائج ستكون أفضل لو استخدمنا اختبار الزوجية ثنائي البعد. في هذا النوع من الرماز يجري توزيع كلمات المعطيات ضمن جدول (أسطر وأعمدة).



- تصميم اختبار زوجية ثنائية الأبعاد -



تأثير خطأ على طرفين



تأثير ثلاثة أخطاء على أربعة أطراف

أربعة أخطاء غير قابلة للاكتشاف

يتم هنا حساب الزوجية لكل سطر ولكل عمود بافتراض أن كلمة المعطيات مكونة من 7 بتات وموضوعة ضمن أسطر مستقلة. يتم إضافة بت زوجية لكل سطر ولكل عمود ويتم بعدها إرسال الجدول بكامله إلى المستقبل الذي يستطيع اكتشاف حتى 3 أخطاء.

2.1.4 رمازات هامنك

تسمح هذه الرمازات بتصحيح الأخطاء عن طريق استخدام مخطط رمازات يحقـ ق $d_{min} = 3$ أي أنه قادر على اكتشاف حتى خطأين وتصحيح خطأ واحد. لنبدأ بحساب العلاقة بين n و k وفق رماز هامنك. نحتاج إلى اختيار عدد طبيعي $m \geq 3$. نستطيع أن نستنتج من m قيمة n والتي تساوي

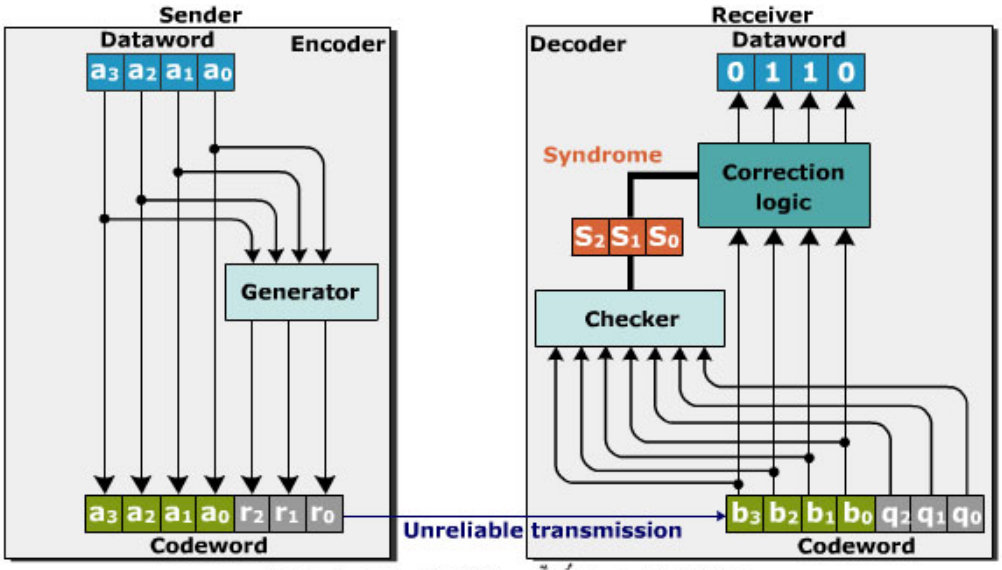
$n = 2^m - 1$ و $k = n - m$.

مثال9:

لنأخذ $m = 3$ مما يؤدي إلى كون $n = 7$. هذا يعطينا رماز هامنك C(7, 4) حيث $d_{min} = 3$. راجع الجدول 4 لمعرفة كلمات المعطيات وكلمات الرماز لهذا الرماز.

Datawords	Codewords	Datawords	Codewords
0000	0000**000**	1000	1000**110**
0001	0001**101**	1001	1001**011**
0010	0010**111**	1010	1010**001**
0011	0011**010**	1011	1011**100**
0100	0100**011**	1100	1100**101**
0101	0101**110**	1101	1101**000**
0110	0110**100**	1110	1110**010**
0111	0111**001**	1111	1111**111**

الجدول 4 – رماز هامنك C(7, 4)



الشكل 10- بنية المُرمِّز ومفكك الرماز لرماز هامنك

لاحظ أن المولد Generator يأخذ نسخة من البتات الأربعة a_0, a_1, a_2, a_3 ويُنشأ 3 بتات زوجية هي r_0, r_1, r_2 كما هو موضح فيما يلي:

$$r_0 = a_2 + a_1 + a_0 \quad \text{modulo-2}$$
$$r_1 = a_3 + a_2 + a_1 \quad \text{modulo-2}$$
$$r_2 = a_1 + a_0 + a_3 \quad \text{modulo-2}$$

يقوم الفاحص checker لدى المستقبل بتوليد مجموعة تدعى Syndrome مكونة من 3 بتات وهي $(s_2s_1s_0)$ حيث:

$$s_0 = b_2 + b_1 + b_0 + q_0 \quad \text{modulo-2}$$
$$s_1 = b_3 + b_2 + b_1 + q_1 \quad \text{modulo-2}$$
$$s_2 = b_1 + b_0 + b_3 + q_2 \quad \text{modulo-2}$$

تسمح البتات الثلاثة $(s_2s_1s_0)$ بقيمها المختلفة (بين 000 و 111) لنا بمعرفة حالات عدم وجود أخطاء أو وجود خطأ في بت محدد حسب الجدول 5.

Syndrome($s_2s_1s_0$)	000	001	010	011	100	101	110	111
Error	None	q_0	q_1	b_2	q_2	b_0	b_3	b_1

الجدول 5 – منطق التصحيح Correction logic المستخدم لدى المستقبل

أي أنه إذا كانت نتيجة الفاحص $s_2s_1s_0 = 0$ هذا يعني أنه لا توجد أخطاء في النقل أما إذا كانت $s_2s_1s_0 = 001$ فهذا يعني أن يوجد خطأ في البت q_0 لكنه لا يهم بالنسبة للمستقبل لأنه بالأصل بت تكراري، أما إذا كانت $s_2s_1s_0 = 011$ فهذا يعني أنه يوجد خطأ في البت b_2 مما يستوجب قلب قيمة هذا البت.

مثال10:

لنتتبع أيضاً مسار 3 كلمات معطيات من المرسل إلى المستقبل:

1. تحول كلمة المعطيات 0100 إلى كلمة الرماز 0100011. عندما يتم استقبال كلمة الرماز 0100011 يكون syndrome = 000 مما يعني أنه لا يوجد أي خطأ في الإرسال وتصبح كلمة المعطيات النهائية هي 0100.
2. تحول كلمة المعطيات 0111 إلى كلمة الرماز 0111001. يتم استقبال كلمة الرماز 0011001. ويكون syndrome = 011. بالرجوع إلى الجدول 5، سيكون لدينا خطأ في b_2. بعد قلب قيمة b_2 (تغيير 0 إلى 1)، تصبح كلمة المعطيات النهائية 0111.
3. تحول كلمة المعطيات 1101 إلى كلمة الرماز 1101000. يتم استقبال كلمة الرماز 00010001 التي تحوي خطأين. تكون قيمة syndrome = 101 مما يعني أن b_0 خطأ. بعد قلب قيمة البت b_0 سنحصل على 0000 أي كلمة المعطيات الخطأ. مما يعني أن الرماز المستخدم لا يستطيع تصحيح خطأين.

5. الرمازات الدوارة Cyclic Codes

تعتبر الرمازات الدوارة حالة خاصة من الرماز الكتلي الخطي مع وجود خاصة واحدة. إذا أزيحت (دُورت) كلمة رماز سنحصل على كلمة رماز أخرى. فإزاحة كلمة الرماز 1011000 دائرياً إلى اليسار تنتج عنه الكلمة 0110001 والتي هي أيضاً كلمة رماز. فإذا اعتبرنا أن كلمة الرماز الأصلية هي a_0 to a_6 والكلمة الناتجة عن الإزاحة الدائرية b_0 to b_6 يتم تغيير البتات وفق الطريقة التالية:

$b_1 = a_0$ $b_2 = a_1$ $b_3 = a_2$ $b_4 = a_3$ $b_5 = a_4$ $b_6 = a_5$

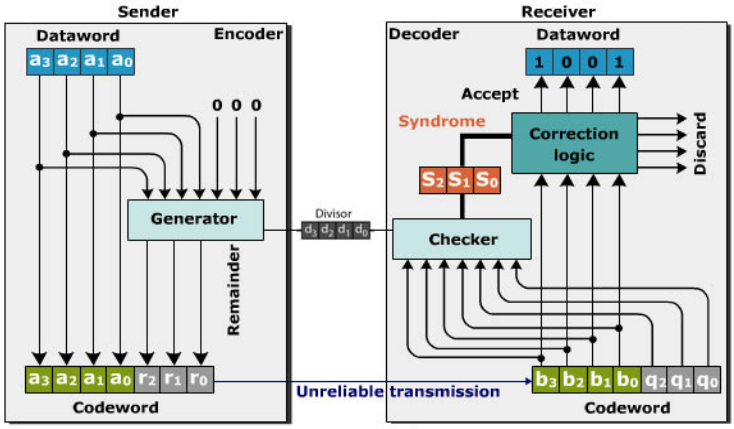
1.5 اختبار التكرارية الدوار Cyclic Redundancy Check (CRC)

يستخدم اختبار التكرارية الدوارة في الشبكات المحلية والواسعة. يبين الجدول 6 مثالاً عن رماز دوار CRC.

Dataword	Codeword	Dataword	Codeword
0000	0000**000**	1000	1000**101**
0001	0001**011**	1001	1001**110**
0010	0010**110**	1010	1010**011**
0011	0011**101**	1011	1011**000**
0100	0100**111**	1100	1100**010**
0101	0101**100**	1101	1101**001**
0110	0110**001**	1110	1110**100**
0111	0111**010**	1111	1111**111**

الجدول 6 – رماز CRC من الشكل C(7, 4)

يبين الشكل التالي كيفية تصميم المُرمز ومفكك الترميز.



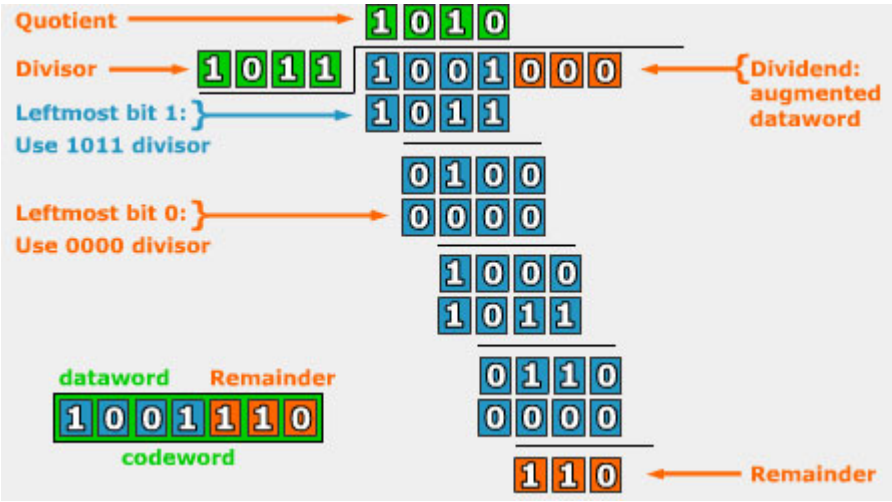
الشكل 12- مُرمِّز ومفكك ترميز CRC

تتألف كلمة المعطيات عند المرمز من k bits (4 في حالتنا هذه)؛ أما كلمة الترميز فهي مؤلفة من n bits (7 في حالتنا هذه). تتم زيادة n-k bits أي 3 بتات ذات القيمة 0 إلى الطرف اليميني من كلمة المعطيات. ندخل الناتج (n bits) إلى المولد generator. يستخدم المولد قاسم divisor متفق عليه مسبقاً بطول $n - k + 1$ (أي 4 بتات هنا).يقوم المولد بتقسيم كلمة الترميز الموسعة على القاسم (قسمة مودولو 2)؛ نهمل حاصل القسمة ونضيف باقي القسمة $(r_2r_1r_0)$ reminder إلى كلمة المعطيات لتكوين كلمة الترميز.

يستقبل مفكك الترميز كلمة المعطيات التي يمكن أن تكون محرفة. ندخل نسخة عن n bits إلى الفاحص الذي هو صورة طبق الأصل عن المولد. يحسب الفاحص باقي القسمة المكون من $n - k$ bits (3 هنا) وتنقل النتيجة إلى محال القرار المنطقي. تتمثل وظيفة محال القرار المنطقي بقبول البتات الأربعة اليسارية من كلمة الترميز إذا كانت syndrome bits are all 0s (أي $S_2S_1S_0 = 000$) وإلا فإنه يتم إهمال كلمة الترميز.

2.5 المرمز Encoder

يبين الشكل التالي طريقة القسمة الثنائية (مودولو 2) التي يقوم بها المرمز.



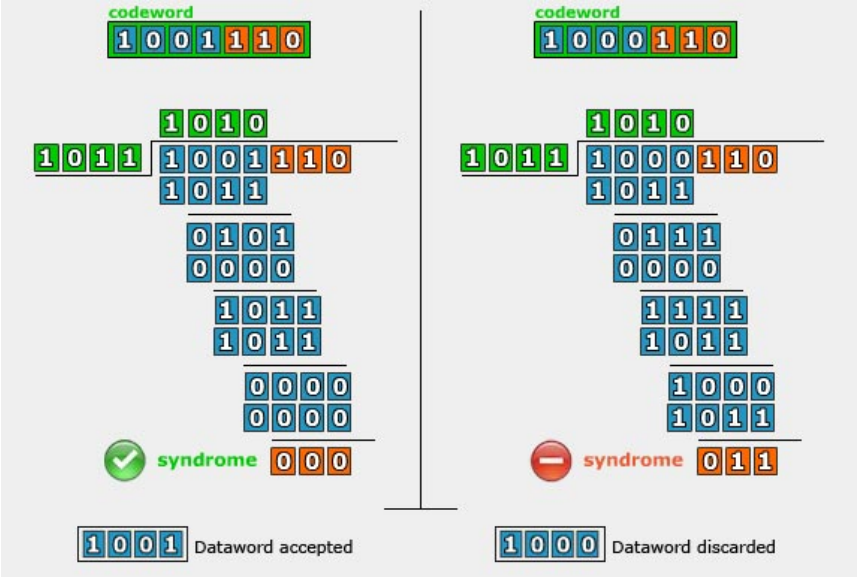
لاحظ هنا أن كلمة المعطيات الأصلية هي 1001 التي يقوم المرمز بإضافة (3 بتات) 000 إليها وبتقسيم الناتج 1001000 على القاسم المتفق عليه 1011 والذي يتألف من 4 بتات. تتم القسم الثنائية مودولو-2 كما هو عليه الحال في القسمة العشرية لكن بما أن الجمع والطرح متكافئين فإننا نستخدم XOR لتحقيقهما.

في كل خطوة من القسمة نأخذ نسخة من القاسم ونطبق عليها XOR مع المقسوم. تستخدم نتيجة XOR (3 بتات في حالتنا هذه) ضمن الخطوة التالية بعض سحب بت واحد لجعل طوله 4 بتات. هناك نقطة أساسية يجب تذكرها في هذا النوع من القسمة: إذا كان البت اليساري من المقسوم 0 فإننا نستخدم قاسم صفرياً all-0s divisor.

عندما لا يعود هناك بتات تسحب إلى الأسفل، نحصل على نتيجة القسمة. تشكل بتات باقي القسمة (r_2, r_1, r_0) بتات الاختبار حيث تتم إضافتها إلى كلمة المعطيات لتكوين كلمة الترميز.

3.5 مفكك الترميز Decoder:

بما أن كلمة الترميز يمكن أن تتغير أثناء الإرسال، فإن مفكك الترميز يقوم بنفس عملية القسمة مثله مثل المرمز. ويكون باقي عملية القيمة هو ما يعرف بالأعراض Syndrome. إذا كان Syndrome = 0 فلا يوجد أخطاء ويتم فصل كلمة المعطيات عن كلمة الترميز وقبولها. إذا كان Syndrome لا يساوي الصفر فتهمل كلمة الترميز وكلمة المعطيات. يبين الشكل 10 حالتين: القسم اليساري يوضح عدم وجود أخطاء في النقل والقسم اليميني يوضح حالة وجود خطأ واحد حيث يكون باقي القسمة مختلف عن الصفر.



يتم عادةً استخدام كثيرات الحدود للدلالة على القاسم. فمثلاً كثير الحدود التالي:

$$X^6 + x + 1$$

يدل على القاسم 1000011

أي كأننا نستخدم كثير الحدود: $1x^6 + 0x^5 + 0x^4 + 0x^3 + 0x^2 + 1x + 1$

6. تدقيق المجموع Checksum

يستخدم تدقيق المجموع في الإنترنت ضمن عدة بروتوكولات لكن ليس على مستوى طبقة وصلة المعطيات. يعتمد تدقيق المجموع على فكرة التكرارية أيضاً.

مثال11:

لنفترض أننا نريد إرسال قائمة مكونة من 5 أرقام كل منها يحوي 4 بتات إلى وجهة ما. فإضافةً إلى إرسال هذه الأرقام، نقوم بإرسال مجموع الأرقام. فإذا كانت القائمة مؤلفة من الأرقام (7, 11, 12, 0, 6) فإننا نرسل (7, 11, 12, 0, 6, 36) حيث يمثل العدد 36 مجموع أرقام القائمة. يقوم المستقبل بجمع الأرقام ومقارنة حاصل الجمع مع المجموع المرسل. إذا تطابق العددان فيعتبر المستقبل أنه لا توجد أخطاء في النقل إما إذا وجد المستقبل فرقاً بين المجموعين فإنه يهمل المعطيات المستقبلة ويعتبر وجود خطأ في الإرسال.

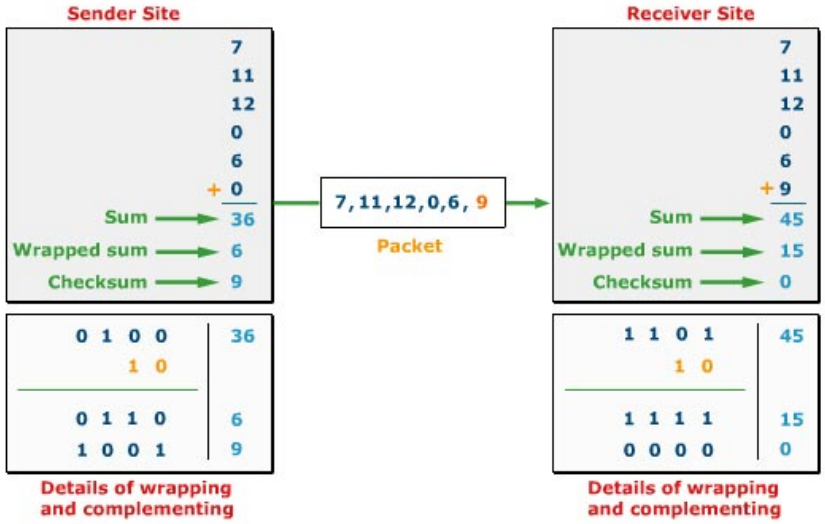
يمكننا أن نسهل عملية المستقبل أكثر لو أننا أرسلنا القيمة -36 السالبة (أو المتممة). فما على المستقبل سوى جمع جميع الأرقام وإذا كانت نتيجة الجمع صفراً فإنه يقبل المعطيات وإلى فيهملها.

لاحظ أن جميع الأرقام المستخدمة سابقاً مؤلفة من 4 بتات ما عدا المجموع . لذلك نستطيع استخدام ما يعرف بالمتمم إلى واحد One complement حيث يتم حساب الأعداد السالبة المتممة عن طريق قلب جميع بتات العدد الموجب.

فمثلاً العدد 21 يمكن تمثيله ثنائياً بالقيمة 10101 لكن بما أن التمثيل الثنائي مؤلف من 4 بتات فإننا نقوم بحذف البت الأخير (إلى اليسار) وجمعه مع باقي البتات فتكون النتيجة : (1 + 0101 =0110 (6)).

مثال12:

لنعيد المثال 11 لكن باستخدام المتمم إلى واحد. يبين الشكل التالي العملية عند المرسل وعند المستقبل. المرسل يقلع بقيمة المجموع مساوية للصفر ويجمع جميع الأرقام مع المجموع الأولي (الذي يعتبر كأنه معطيات). تصبح النتيجة 36. بما أن $100100_2 = 36_{10}$ فإننا نحذف آخر بتين من اليسار ونجمعهما مع الباقي، أي 0110 (6) = 0100 + 10. نتمم المجموع فيصبح 1001 أي 9 ونرسل الرقم 9 على أنه تدقيق المجموع Checksum.عندها يقوم المستقبل بالعملية نفسها أي بجمع الأرقام بما فيها تدقيق المجموع فسيحصل على القيمة 0 إذا لم يكن هناك أخطاء في النقل. إذا كانت نتيجة الجمع لا تساوي 0 فإن المستقبل يهمل الأرقام أو المعطيات المستقبلة.



الوحدة الرابعة

التحكم ضمن طبقة وصلة المعطيات

Data Link Control

تهتم طبقة وصلة المعطيات بالنقل من عقدة إلى عقدة. لتحقيق ذلك تقوم بمجموعة من الوظائف مثل التأطير Framing والتحكم بالأخطاء والتدفق وتحقيق البروتوكولات التي تؤمن نقل الأطر بين العقد المتجاورة نقلاً موثوقاً.

1. التأطير Framing

تقوم الطبقة الفيزيائية بنقل البتات على شكل إشارات من المصدر إلى الوجهة. هذه البتات تحتاج إلى تجميع ضمن أطر بشكل يسمح بالتمييز بين هذه الأطر. فالنظام البريدي، على سبيل المثال، يسمح بالتمييز بين الرسائل المختلفة عن طريق وضع كل رسالة داخل مغلف مستقل.

فالتأطير على مستوى طبقة وصلة المعطيات يسمح بتمييز رسالة من مصدر محدد إلى وجهة محددة أو بالتمييز بين الرسالة وبين رسائل أخرى موجهة إلى وجهات مختلفة، وذلك عن طريق إضافة عنواني المصدر والوجهة.

يفيد عنوان الوجهة بمعرفة إلى أين ستذهب الرسالة بينما تستفيد الوجهة من عنوان المصدر في حال أرادت إقرار الاستلام.

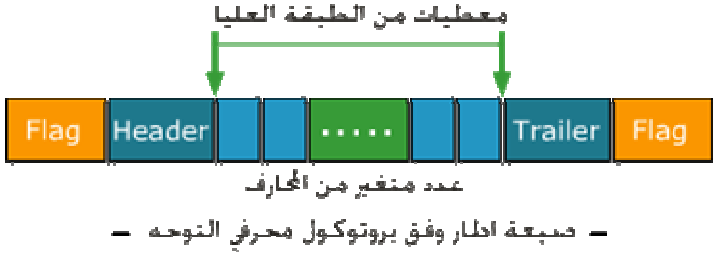
مع أنه من الممكن إرسال كامل الرسالة ضمن إطار واحد إلا أن ذلك غير مقبول عملياً، فتحريف أي بت من الرسالة، الذي يزداد احتمال وقوعه مع ازدياد طول الرسالة، يقتضي إعادة إرسال كل الرسالة من جديد، أضف إلى ذلك عدم فعالية التحكم بالتدفق وبالأخطاء.

الأطر ذات الطول الثابت Fixed-size framing

يمكن أن يكون طول الإطار ثابت أو متغير. الأطر ثابتة الطول لا تتطلب تحديد حدوداً للإطار بينما الأطر ذات الطول المتغير المستخدمة في الشبكات المحلية فإنها تحتاج إلى تعريف نهاية كل إطار وبداية الإطار التالي. لقد جرى، تاريخياً، استخدام طريقتين لتحقيق ذلك : طريقة محرفية التوجه Character-oriented وطريقة بتية التوجه Bit-oriented.

1. البروتوكولات محرفية التوجه Character-oriented protocols

تكون المعطيات، وفق هذا النوع من البروتوكولات، عبارة عن محارف مؤلفة من ثمانية بتات تستخدم أحد أنظمة الترميز المعروفة مثل نظام ASCII. تكون الترويسة Header واللاحقة Trailer من مضاعفات 8-bits. نضيف راية Flag مؤلفة من بايت واحد 8-bits للدلالة على بداية الإطار وراية أخرى للدلالة على نهايته؛ تنفع هذه الرايات في الفصل بين الأطر.

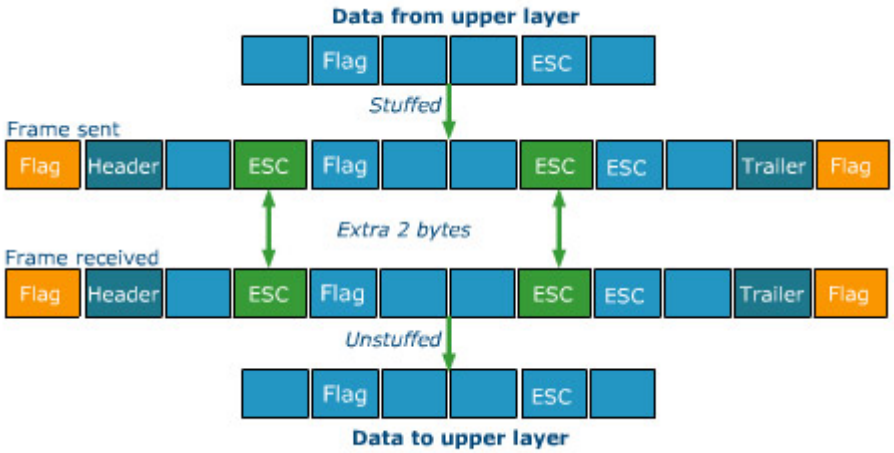


يصلح هذا النوع من البروتوكولات عند نقل النصوص فقط لأنه في هذه الحالة يمكننا اختيار محرف غير مستخدم ضمن النص للدلالة على الراية. لكن مع تنوع طبيعة المعطيات المتناقلة حالياً مثل الصور والفيديو والصوت، أصبح من غير الممكن ضمان عدم وجود الراية ضمن المعطيات. فإذا حدث ذلك فإن المستقبل عندما يقرأ حقل الراية في منتصف المعطيات فإنه يعتقد أنه وصل إلى نهاية الإطار. للتخلص من هذه المشكلة يتم استخدام طريقة حشو البايتات Byte-stuffing.

1.1 طريقة حشو البايتات Byte-stuffing

يتم إضافة بايت خاص إلى جزء الإطار المخصص للمعطيات فقط عندما يوجد محرف الراية. يدعى هذا المحرف عادةٍ بمحرف الإفلات ESC (Escape character). عندما يصادف المستقبل محرف الإفلات فإنه يحذفه ويعالج المحرف التالي على أنه محرف معطيات وليس محرف بداية أو نهاية الإطار.

إضافةً إلى ما سبق، فإنه يتم إضافة محرف الإفلات ESC عند وجود محرف ESC ضمن المعطيات نفسها كما هو موضح في الشكل التالي:

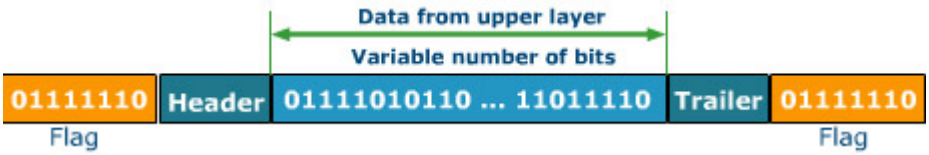


حشو البايتات وإزالة الحشو

تبقى لدينا مشكلة أخرى مع البروتوكولات محرفية التوجه أن بعض أنظمة الترميز الحديثة مثل Unicode تستخدم محارف ذات طول bit-16 أو bit-32 الأمر الذي يتعارض مع المحارف الثمانية.

2. البروتوكولات بتية التوجه Bit-oriented Protocols

يجري هنا ترميز جزء المعطيات من الإطار بسلسلة بتات التي يمكن أن تمثل الصوت والصورة والنصوص. تبقى الحاجة موجودة للفصل بين الأطر. تستخدم معظم البروتوكولات راية مكونة من ثمانية بتات **01111110** للدلالة على بداية أو نهاية كل إطار كما هو موضح في الشكل.



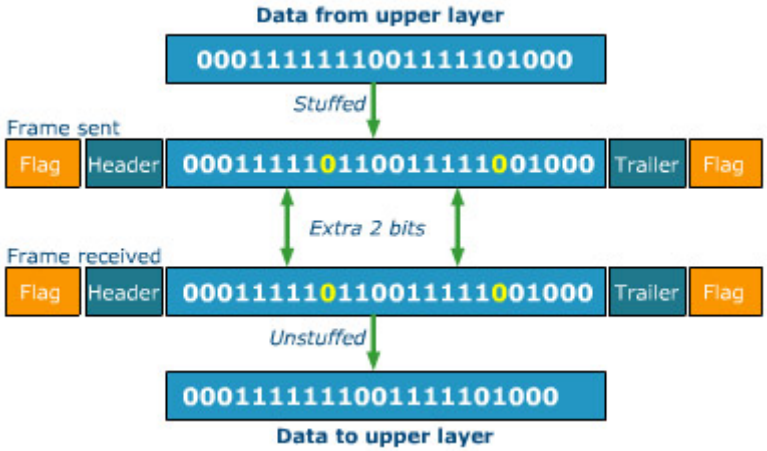
إطار وفق بروتوكول بتي التوجه

حتى نضمن عدم وجود راية البداية أو النهاية ضمن حقل المعطيات فإننا نقوم بتنفيذ طريقة الحشو البتي Bit stuffing.

1.2 طريق الحشو البتي Bit-stuffing

عندما يتم مصادفة صفر يليه خمس واحدات فإن المرسل يضيف صفر آخر بعدها. طبعاً يقوم المستقبل بحذف هذا البت الإضافي من المعطيات.

ملاحظة: تجري إضافة الصفر بعد مصادفة صفر تليه خمس واحدات بغض النظر عن البت التالي. يبين الشكل التالي كيفية إضافة البت لدى المرسل وكيفية حذفه لدى المستقبل.



حشو البتات وإزالة الحشو

2. بروتوكولات التحكم بالتدفق وبالأخطاء

التحكم بالتدفق

يعتبر التحكم بالتدفق من أهم واجبات طبقة وصلة المعطيات حيث يسعى إلى تحديد كمية المعطيات التي يستطيع المرسل إرسالها قبل استلام إقرار من المستقبل. الهدف الأساسي للتحكم بالتدفق هو عدم إغراق المستقبل بمعطيات غير قادر على معالجتها بسبب سرعته المحدودة أو حجم الذاكرة المتوفرة له. لذلك يجب أن يستطيع المستقبل إعلام المرسل قبل الوصول إلى حدوده القصوى ليبطئ إرساله أو ليوقفه لفترة مؤقتة. بما أن سرعة المعالجة بشكل عام أقل من سرعة نقل المعطيات فيجب أن يتوفر لدى المستقبل ذاكرة دارئة Buffer يخزن فيها المعطيات قبيل معالجتها. فعندما تصبح الذاكرة شبه ممتلئة، يجب أن يكون باستطاعة المستقبل إعلام المرسل لتخفيف سرعة الإرسال.

تعريف: التحكم بالتدفق هو مجموعة الإجرائيات المستخدمة لتحديد كمية المعطيات التي يستطيع المرسل إرسالها قبل انتظار استلام إقرار من المستقبل.

التحكم بالأخطاء

يشمل التحكم بالأخطاء بشكل عام على آليات اكتشاف أخطاء النقل وتصحيحها. فهو يسمح للمستقبل بإعلام المرسل عن الأطر الضائعة أو المحرفة أثناء النقل وينسق آلية إعادة نقل هذه الأطر مع المرسل. أما على مستوى طبقة وصلة المعطيات فيتم التصحيح عن طريق إعادة النقل أو ما يعرف بطلب الإعادة الآلي ARQ (Automatic repeat request).

البروتوكولات

تفيد البروتوكولات في ضم وظائف التأطير والتحكم بالتدفق والأخطاء معاً لتحقيق النقل السليم للمعطيات من عقدة إلى أخرى. يتم تحقيق البروتوكول باستخدام لغة برمجة ما.

سنقسم فيما يلي البروتوكولات إلى نوعين: النوع الأول وهو نظري ولا يوجد في الحياة العملية إذ يعتمد على كون قناة الاتصال مثالية بينما يعتمد النوع الثاني على كون قناة الاتصال غير مثالية.

1.2 بروتوكولات القناة المثالية

تعاريف:

القناة المثالية: هي القناة التي لا تضيع الأطر ولا تحرفها ولا تكررها.

تحريف الأطر: تغيير بت أو أكثر في الإطار المرسل.

تكرار الأطر : وصول نسختين من نفس الإطار.

1.1.2 البروتوكول المبسط

يتميز هذا البروتوكول بكونه لا يمتلك أية آلية للتعامل مع أخطاء النقل أو مع التدفق وبكونه وحيد الاتجاه. نفترض أيضاً أن المستقبل يستطيع معالجة أي إطار مستقبل بزمن معالجة مهمل. أي أن طبقة وصلة المعطيات عند المرسل تستلم المعطيات من طبقة الشبكة ومن ثم تبني الإطار وترسله بينما تقوم طبقة وصلة المعطيات عند المستقبل باستقبال الإطار القادم من الطبقة الفيزيائية وتستخلص منه المعطيات وتسلمها إلى طبقة الشبكة.

لاحظ أن المرسل لا يستطيع الإرسال إذا لم يستلم طرد معطيات من طبقة الشبكة وكذلك الأمر بالنسبة للمستقبل الذي لا يستطيع تسليم طرد المعطيات إلى طبقة الشبكة إذا لم يستقبل إطار معطيات من الطبقة الفيزيائية. أو بشكل آخر، إذا قمنا بتحقيق البروتوكول عن طريق إجرائيات Procedures فإننا نحتاج لإدخال مفهوم الحدث Event داخل البروتوكول. تُنفذ الإجرائية لدى المرسل بشكل مستمر وتكون بحالة انتظار دائم بحيث أنه لا يوجد أي فعل إلا عندما يصل طلب من طبقة الشبكة. كذلك الأمر بالنسبة للإجرائية لدى المستقبل التي تكون بحالة انتظار ورود إعلام Notification من الطبقة الفيزيائية بوصول إطار. أي أن الإجرائيتين تكونان في حالة عمل مستمر لأنهما لا يعرفان بشكل مسبق لحظة وقوع الحدث.

```
1   While(true)                    // تكرار إلى اللانهاية
2   {
3    WaitForEvent();               // انتظار وقوع حدث
4    If(Event(RequestToSend))      //يوجد طرد للإرسال
5    {
6     GetData();                   //استلام طرد معطيات من طبقة الشبكة
7     MakeFrame();                 //إضافة الترويسة واللاحقة
8     SendData();                  //إرسال الإطار إلى الطبقة الفيزيائية
9    }
10  }
```

خوارزمية 1- الإجرائية التي يقوم بها المرسل

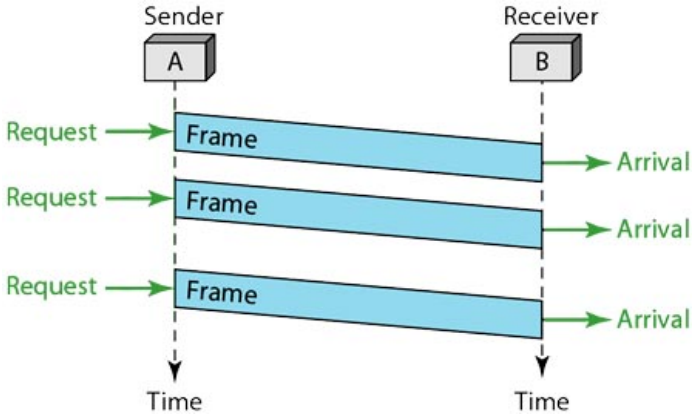
لاحظ هنا أن المرسل يدخل في حلقة مغلقة (السطر 3 حتى 9) وفي كل مرة يقع حدث (السطر 4) فإنه يعالج الطلب ويعود من جديد إلى حالة الانتظار.

```
1  While(true)                              //  تكرار إلى اللانهاية
2  {
3      WaitForEvent();                      //  انتظار وقوع حدث
4      If(Event(ArrivalNotification))       //وصول إطار معطيات
5      {
6         ReceiveFrame();                   //  استقبال الإطار من الطبقة الفيزيائية
7         ExtractData();                    //  استخلاص طرد المعطيات من الإطار
8         DeliverData();                    //تسليم المعطيات لطبقة الشبكة
9      }
10 }
```

خوارزمية 2 – الإجرائية التي يقوم بها المستقبل

أما المستقبل فيعالج المعطيات تصاعدياً من الطبقة الفيزيائية وصولاً إلى طبقة الشبكة.

يبين الشكل التالي المخطط الزمني لتنفيذ هذا البروتوكول.



المخطط الزمني Time Sequence Diagram للبروتوكول المبسط

2.1.2 بروتوكول التوقف والانتظار Stop-and-wait

يهدف هذا البروتوكول إلى التحكم بالتدفق بين المرسل والمستقبل حتى لا تمتلئ الذاكرة لدى المستقبل ويبدأ بإهمال المعطيات الجديدة.

ولتحقيق ذلك يرسل المرسل إطار واحد ويتوقف حتى استلام تأكيد من المستقبل (جاهزية المستقبل لمتابعة الاستقبال) ومن ثم يرسل الإطار التالي وهكذا.

لاحظ أن المعطيات تتدفق باتجاه واحد (من المرسل إلى المستقبل) لكننا نحتاج أيضاً إلى إطارات تحكم من المستقبل إلى المرسل مثل Ack frame.

لنرى كيف ستصبح إجرائيات المرسل والمستقبل.

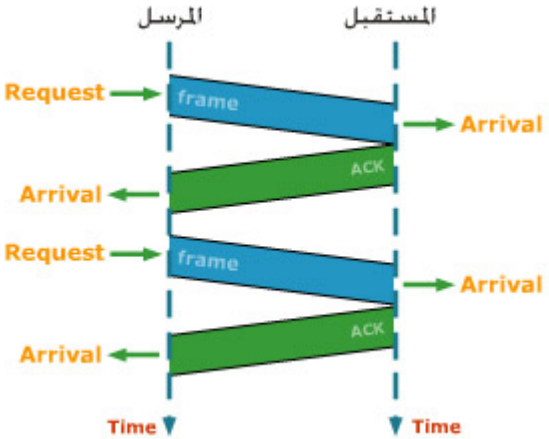
```
1  While(true)                                  // تكرار إلى اللانهاية
2  CanSend = true;                              //السماح بإرسال أول إطار
3  {
4    WaitForEvent();                            // انتظار وقوع حدث
5    If(Event(RequestToSend)AND CanSend)        //يوجد طرد للإرسال ويسمح بالإرسال
6    {
7     GetData();                                //استقبال الطرد من طبقة الشبكة
8     MakeFrame();                              // بناء الإطار
9     SendFrame();                              // إرسال الإطار إلى الطبقة الفيزيائية
10    CanSend = false;                          //توقف الإرسال حتى استلام تأكيد من المستقبل
11   }
12   WaitForEvent();
13   If (Event(ArrivalNotification))           //وصول التأكيد من المستقبل
14   {
15    ReceiveFrame();                           //استلام الإطار الذي يحوي التأكيد
16    CanSend = true;                           //يمكننا الآن إرسال إطار جديد
17   }
18 }
```

خوارزمية 3 – إجرائية المرسل في بروتوكول التوقف والانتظار

```
1  While(true)                                  // تكرار إلى اللانهاية
2  {
3    WaitForEvent();                            // انتظار وقوع حدث
4    If(Event(ArrivalNotification))             //وصول إطار معطيات
5    {
6     ReceiveFrame();                           // استقبال الإطار من الطبقة الفيزيائية
7     ExtractData();                            // استخلاص طرد المعطيات من الإطار
8     DeliverData();                            //تسليم المعطيات لطبقة الشبكة
9     SendFrame();                              // إرسال إطار تأكيد
10   }
11 }
```

خوارزمية 4 – إجرائية المستقبل في بروتوكول التوقف والانتظار

يبين الشكل التالي المخطط الزمني للاتصال بين المرسل والمستقبل:



المخطط الزمني لبروتوكول الإرسال والتوقف

2.2 بروتوكولات القناة غير المثالية

سنعالج الآن قضية كون القناة غير مثالية لذلك ينبغي أن نضيف إمكانية التحكم بالأخطاء للبروتوكول المبسط السابق. لذلك، سنناقش ثلاثة أنواع من البروتوكولات.

1.2.2 التوقف والانتظار مع طلب الإعادة الآلي Stop-and-Wait Automatic Repeat Request

نفترض أولاً أن المرسل يضيف بتات تكرارية تسمح للمستقبل باكتشاف الأخطاء. فعندما يستقبل المستقبل إطاراً ما فإنه يختبر كونه خالٍ من الأخطاء، فإذا كان يحوي خطأ ما فإنه يهمله بصمت.

تبقى مشكلة الإطارات الضائعة أو المكررة أو المستلمة خارج الترتيب.

عندما يتم إرسال عدة إطارات وفق ترتيب ما فيجب أن يتم استقبالها وفق نفس الترتيب وفي حال استلام إطار لا يحترم الترتيب فإننا نقول أن الإطار المستلم خارج الترتيب Out of order.

لا بد هنا من ترقيم الإطارات بطريقة تسمح للمستقبل بتحديد كون إطار مستلم ما خارج الترتيب، الأمر الذي يدل على ضياع إطارات أخرى أو تكرارها على سبيل المثال.

يجب أيضاً إعادة إرسال الأطر الفاسدة أو الضائعة الأمر الذي يحتم على المرسل إبقاء نسخة من كل إطار مرسل وفي كل مرة يرسل فيها إطار ما فإنه يقلع مؤقت زمني Timer . إذا انقضى الوقت دون استلام إقرار تأكيد Ack من المستقبل فإن المرسل يعيد إرسال النسخة المخزنة عنده ويعيد إقلاع المؤقت الزمني.

بما أن إطار إقرار التأكيد Ack frame يمكن أن يضيع أيضاً أو يفسد أثناء الإرسال فيجب إضافة بتات تكرارية له ورقم تسلسلي. بناء على ما سبق، يقوم المرسل بإهمال إطار الإقرار إذا كان فاسداً أو إذا كان خارج الترتيب.

<u>الأرقام التسلسلية Sequence Numbers</u>

تتم إضافة الأرقام التسلسلية إلى أطر المعطيات عن طريق حقل خاص بالترقيم. يجب التفكير هنا بطول هذا الحقل لأنه لا يمكن أن يصل الرقم التسلسلي إلى اللانهاية وإنما له حد أعلى . فإذا

خصصنا m bits للدلالة على الرقم التسلسلي فإننا بحاجة لإعادة الأرقام التسلسلية إلى الصفر بعد القيمة $2^m - 1$.

نحتاج، ضمن بروتوكول التوقف والانتظار مع إعادة الطلب الآلي، إلى أرقام تسلسلية مودولو-2 (0 or 1).

أرقام الإقرارات Acknowledgment Numbers

يعلن كل إقرار عن الرقم التسلسلي مودولو 2 للإطار التالي المنتظر. أي أنه إذا استقبل المستقبل إطار المعطيات رقم 0 بشكل سليم دون أخطاء فإنه سيرسل إطار إقرار يحمل الرقم 1 للدلالة على أنه ينتظر الإطار التالي وهو ذو الرقم 1.

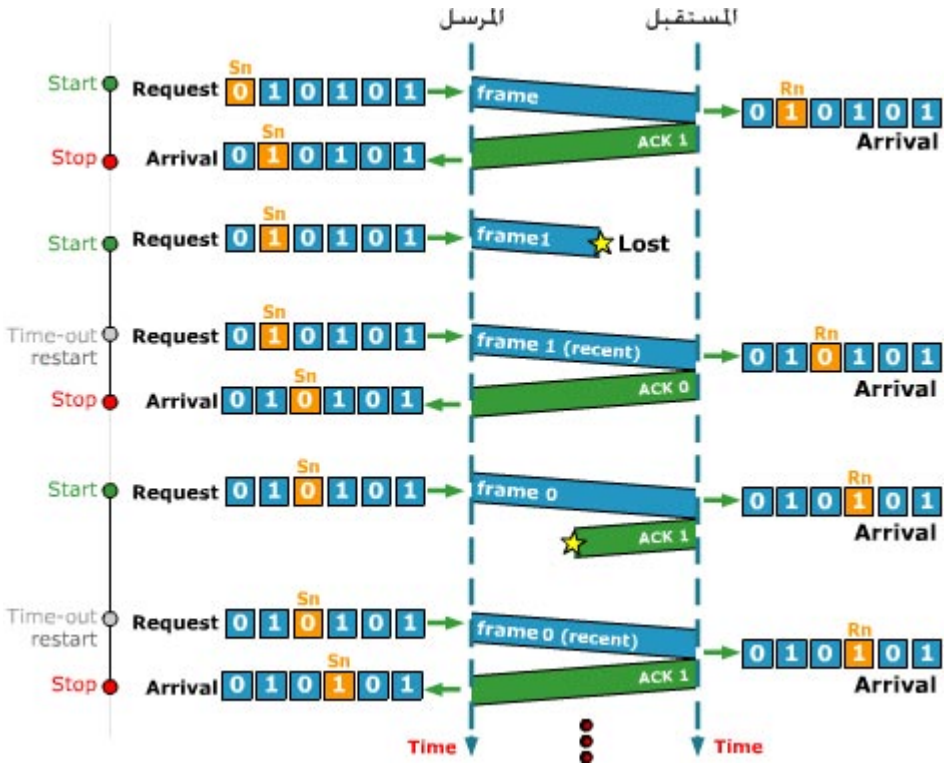
```
1  Sn = 0;                                          // البدء بإرسال الإطار رقم صفر
2  CanSend = true;                                  // السماح بإرسال أول إطار
3  While(true)                                      // تكرار إلى اللانهاية
4  {
5    WaitForEvent();                                // انتظار وقوع حدث
6    If(Event(RequestToSend)AND CanSend)            // يوجد طرد للإرسال ويسمح بالإرسال
7    {
8      GetData();                                   // استقبال الطرد من طبقة الشبكة
9      MakeFrame(Sn);                               // بناء الإطار ذو الرقم Sn
10     StoreFrame(Sn);                              // تخزين الإطار ذو الرقم Sn
11     SendFrame(Sn);                               // إرسال الإطار إلى الطبقة الفيزيائية
12     StartTimer();                                // إقلاع المؤقت الزمني
13     Sn = Sn + 1;                                 // زيادة الرقم التسلسلي
14     CanSend = false;                             // توقف الإرسال حتى استلام تأكيد من المستقبل
15   }
16   WaitForEvent();
17   If (Event(ArrivalNotification))                // وصول التأكيد من المستقبل
18   {
19     ReceiveFrame(AckNo);                         // استلام الإطار الذي يحوي رقم التأكيد
20     If(NOT corrupted AND AckNo == Sn)            // إذا لم يكن الإطار المستلم فاسداً ورقم الإقرار صحيح
21     {
22       StopTimer();                               // توقيف المؤقت الزمني
23       PurgeFrame(Sn-1);                          // حذف النسخة المخزنة من الإطار
24       CanSend = true;
25     }
26   }
27
28     If(Event(TimeOut))                           // انقضاء مدة المؤقت الزمني
29     {
30       StartTimer();
31       ResendFrame(Sn-1);                         // إعادة إرسال النسخة المخزنة من الإطار
32     }
33 }
```

خوارزمية 5 – إجرائية المرسل في بروتوكول التوقف والانتظار مع إعادة الطلب الآلي Stop-and-Wait ARQ

```
1  Rn = 0;                                  //انتظار استقبال الإطار رقم صفر أولاً
2  While(true)                              //  تكرار إلى اللانهاية
3  {
4     WaitForEvent();                       //  انتظار وقوع حدث
5     If(Event(ArrivalNotification))        //وصول إطار معطيات
6     {
7      ReceiveFrame();                      //  استقبال الإطار من الطبقة الفيزيائية
8      if(correptud(frame));                //الإطار فاسد
9       Sleep();                            //انتظار
10     if(seqNo == Rn)                      //الرقم التسلسلي للإطار المستقبل هو نفسه المنتظر
11     {
12      ExtractData();                      //استخلاص المعطيات من الإطار
13      DeliverData();                      //تسليم المعطيات لطبقة الشبكة
14      Rn = Rn + 1;                        //زيادة الرقم التسلسلي المنتظر
15     }
16      SendFrame(Rn);                      //إرسال إطار إقرار يحوي الرقم التسلسلي
17    }
18 }
```

خوارزمية 5 – إجرائية المستقبل في بروتوكول التوقف والانتظار مع إعادة الطلب الآلي Stop-and-Wait ARQ

يبين الشكل التالي كيفية عمل هذا البروتوكول:



المخطط الزمني للبروتوكول السابق

لاحظ من الشكل السابق أنه يتم إرسال الإطار 0 واستلام إقرار بوصوله سالماً بينما يضيع الإطار رقم 1 ويعاد إرساله بعد انقضاء فترة المؤقت الزمني. يجري أيضاً إقرار استلام الإطار رقم 1 المعاد إرساله ويجري إيقاف المؤقت الزمني. يجري إرسال الإطار رقم 0 ويجري إرسال إقرار الاستلام لكن الإقرار يضيع؛ بما أن المرسل لا يستطيع أن يعرف فيما إذا كان الإطار الذي ضاع أو الإقرار فيعيد إرسال الإطار بعد انقضاء فترة المؤقت الزمني حتى يتم استلام الإقرار بشكل سليم.

فعالية البروتوكول

إن بروتوكول التوقف والانتظار مع إعادة الطلب الآلي غير فعال خاصةً عندما تكون قناة الاتصال عريضة وطويلة. نقصد بقناة عريضة إمكانية إرسال عدد كبير من البتات في الثانية ونقصد بقناة طويلة كون تأخير الذهاب والإياب Round-trip delay طويل (أي زمن تأخير البت لتحقيق رحلة ذهاب وإياب من وإلى المرسل) .

تعريف: جداء عرض الحزمة Bandwidth-delay product

هو جداء عرض الحزمة في تأخير الذهاب والإياب = Bandwidth (in bps) x Delay (second)

يعود سبب عدم الفعالية إلى أننا لا نستطيع استخدام عرض الحزمة المتوفر بشكل فعال حيث أن المرسل مضطر إلى انتظار الإقرار حتى يرسل إطار آخر.

مثال 1

افترض أن عرض الحزمة المتوفر هو 1 Mbps وأن كل بت يستغرق 20 ms لتحقيق رحلة ذهاب وإياب. ما هو جداء تأخير الحزمة ؟ وفي حال كان طول إطار المعطيات هو 1000 bits، ما هي نسبة استخدام القناة؟

الحل:

جداء تأخير الحزمة هو:

$$(1 \times 10^{6}) \times (20 \times 10^{-3}) = 20{,}000 \text{ bits}$$

هذا يعني أن النظام يستطيع إرسال 20,000 bits خلال زمن إرسال المعطيات من قبل المرسل ووصولها إلى المستقبل وعودتها من جديد إلى المرسل. لكن نظامنا يرسل فقط 1000 bits مما يعني أن نسبة استخدام القناة هي:

$$1{,}000 / 20{,}000 = 5\ \%$$

تدريب

احسب نسبة استخدام القناة لو كان البروتوكول يرسل 12 إطاراً قبل أن ينتظر وصول إطار الإقرار.

2.2.2 طلب الإعادة الآلي بالعودة ن-خطوة Go-Back-N ARQ

لاحظ من الأمثلة السابقة أننا نستطيع تحسين فعالية النقل إذا سمحنا للمرسل بإرسال عدة أطر قبل انتظار الإقرار وهذا ما يفعله بروتوكول Go-Back-N ARQ.

- **الأرقام التسلسلية Sequence Numbers**

يجري هنا ترقيم الأطر تسلسلياً إلى حد أعلى ومن ثم نعود إلى البداية (0). فإذا افترضنا أن طول حقل الرقم التسلسلي هو m bits فهذا يعني أننا نستطيع الترقيم من القيمة 0 إلى القيمة 2^m-12. أو بشكل آخر إن الأرقام التسلسلية هي مودولو 2^m.

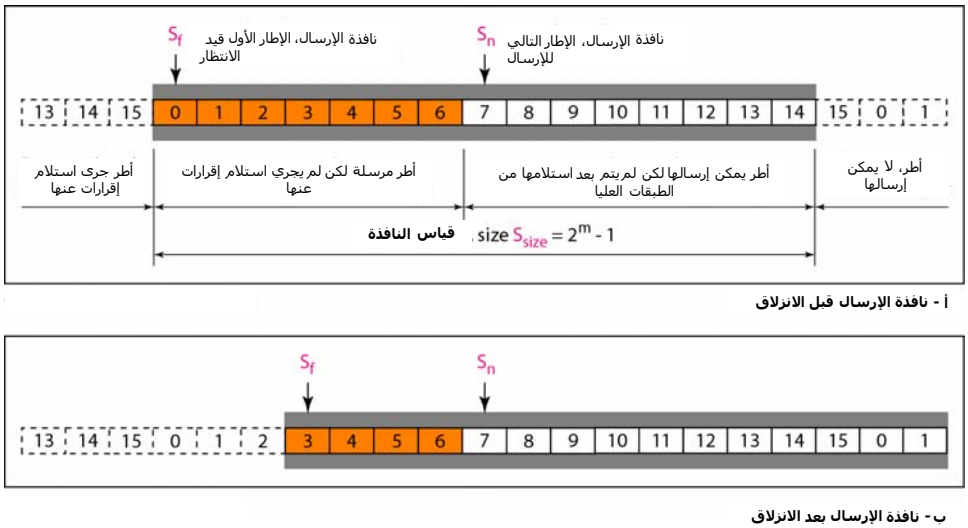
فإذا كان m=3 فإن الأرقام التسلسلية ستكون:

→ 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, … →

- **النافذة المنزلقة Sliding Window**

تحدد النافذة المنزلقة للمرسل (أو للمستقبل) مجال الأرقام التسلسلية التي يستطيع أن يتعامل معها المرسل (أو المستقبل) في لحظة معينة. تحدد نافذة الإرسال الأرقام التسلسلية التي يمكن إرسالها. ضمن مكان ما من النافذة، تشير بعض الأرقام التسلسلية إلى الإطارات التي جرى إرسالها فعلاً ويشير البعض الآخر إلى الأرقام التسلسلية التي يمكن إرسالها. سنعتبر أن قياس النافذة هو ثابت ويحدد الطول الأعظمي لها بالقيمة 2^m-12.

يبين الشكل التالي نافذة منزلقة ذات طول 15 (أي أن m=4).



نافذة الإرسال لبروتوكول Go-Back-N ARQ

تقسم النافذة، في لحظة معينة، الأرقام التسلسلية المحتملة إلى أربع مناطق. المنطقة الأولى، الممتدة من أقصى يسار الشكل وصولاً إلى الجدار اليساري للنافذة، تحدد الأرقام التسلسلية للأطر التي جرى استلام إقرارات عنها. لا يعير المرسل أي اهتمام لهذه الأرقام ولا يحفظ أي نسخ عنها. المنطقة الثانية، الملونة باللون الأحمر من النافذة (تحمل الأرقام من 0 إلى 6)، تحدد مجال الأرقام التسلسلية التابعة لأطر جرى إرسالها ولم يجري استلام أي إقرار عنها. يحتاج المرسل إلى الانتظار ليعرف فيما إذا كانت هذه الأطر أو بعضها وصلت بسلام أو ضاعت في الطريق. المنطقة الثالثة، البيضاء، تحدد مجال الأرقام التسلسلية للأطر التي يمكن إرسالها؛ لكن لم يتم استلامها بعد من الطبقات العليا. تحدد المنطقة الرابعة أرقام الأطر التي لا يمكن إرسالها قبل أن تنزلق النافذة.

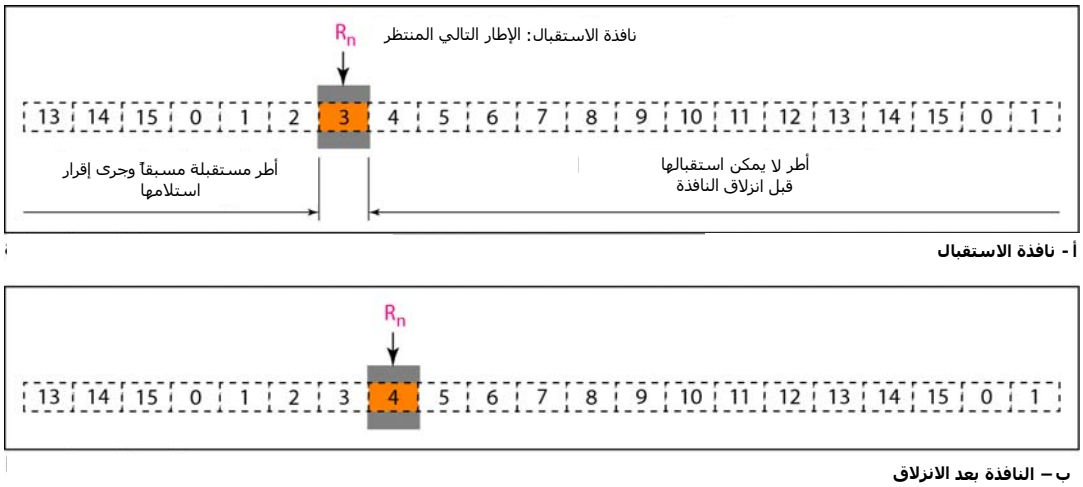
لاحظ أن النافذة، التي هي عبارة عن مفهوم مجرد، يمكن تعريفها باستخدام 3 متحولات وهي قياسها وموضعها في لحظة ما:

1. نافذة الإرسال: الإطار الأول قيد الانتظار Send Window, the first outstanding frame Sf
2. نافذة الإرسال: الإطار التالي للإرسال Send windows, the next frame to be sent Sn
3. نافذة الإرسال: قياس النافذة Send Window, Size Ssize

يبين الشكل 8-ب كيف يمكن للنافذة أن تنزلق خانة واحدة أو أكثر إلى اليمين عندما يجري استلام إقرار من الطرف الأخر. لاحظ هنا أن الإقرارات تراكمية أي أننا نستطيع إقرار استلام أكثر من إطار واحد باستخدام إطار إقرار وحيد One ACK. بالعودة إلى الشكل 8-ب، جرى إقرار استلام الأطر المرقمة 0, 1 and 2 مما أدى إلى انزلاق النافذة 3 خانات. لاحظ أن Sf = 3 لأن الإطار رقم 3 هو الإطار قيد الانتظار حالياً.

بالنسبة لنافذة المستقبل فهي تحاول التأكد من أن الأطر السليمة جرى استلامها وجرى إرسال الإقرارات المناسبة. يكون قياس نافذة المستقبل دوماً 1 ويكون المستقبل دوماً في حالة انتظار إطار ما. عندما يجري استلام إطار خارج الترتيب فإنه يهمله ويجب إعادة إرساله.

يبين الشكل التالي نافذة المستقبل.



نافذة الاستقبال لبروتوكول Go-Back-N ARQ

لاحظ أننا نحتاج هنا إلى متحول واحد وهو الإطار التالي المنتظر (Receive window, next frame expected) Rn . تؤشر الأرقام التسلسلية الواقعة إلى يسار Rn على الأطر المستلمة والمرسل إقرار عنها بينما تؤشر الأطر الواقعة إلى يمين Rn على الأطر التي لا يمكن استلامها. لذلك فإن استلام أي إطار ذو رقم تسلسلي واقع ضمن هاتين المنطقتين يؤدي إلى إهماله ولا يتم قبول وإقرار استلام إلى الإطار الذي يحمل رقم تسلسلي مساوٍ للمتحول Rn.

تنزلق نافذة المستقبل أيضاً لكن خانة واحدة في كل مرة عندما يجري استلام الإطار المنتظر فقط.

- **المؤقت الزمني Timer**

نستخدم في هذا البروتوكول مؤقت زمني وحيد وذلك لأن المؤقت المستخدم لأول إطار جرى إرساله تنقضي مدته أولاً.

- **الإقرارات Acknowledgments**

يرسل المستقبل إقراراً موجباً Positive Ack إذا جرى استقبال إطار سليم وضمن الترتيب وإلا فإن المستقبل يهمل الإطار وجميع الأطر التالية ولا يفعل شيئاً (يصمت) حتى يستقبل الإطار المنتظر. إن صمت المستقبل يؤدي إلى انقضاء الفترة الزمنية المحددة لاستلام إقرار الإطار الأمر الذي يضطر المرسل إلى العودة Go-back وإعادة إرسال جميع الأطر من الإطار الذي انقضى مؤقته الزمني والأطر التالية.

لا يحتاج المستقبل إلى إقرار استلام كل إطار سليم مستقبل وإنما يستطيع إرسال إقرار استلام تراكمي لمجموعة أطر معاً.

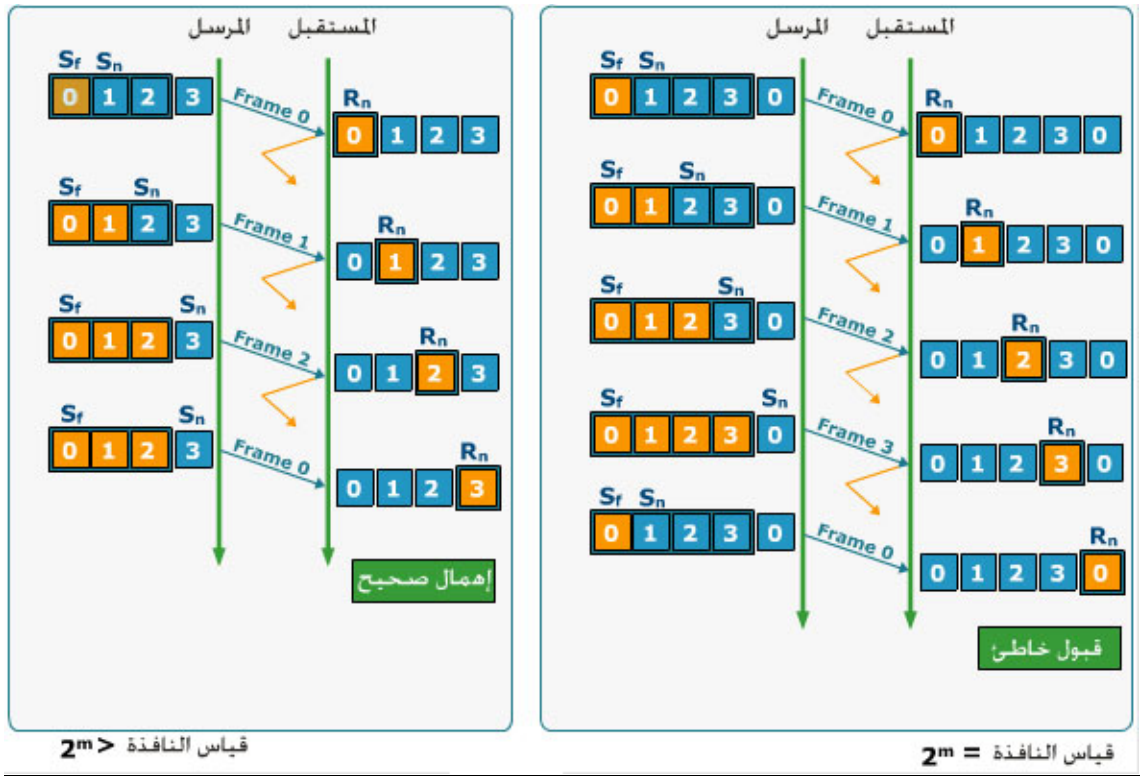
- **إعادة إرسال إطار**

عندما ينقضي المؤقت الزمني فإن المرسل يعيد إرسال جميع الأطر قيد الانتظار. افترض أن المرسل أرسل الإطار رقم 6 لكن المؤقت الزمني المتعلق بالإطار 3 انقضى. الأمر الذي يعني أن الإطار رقم 3 لم يجر إقرار استلامه؛ لذلك يعود المرسل ويرسل الأطر 3, 4, 5, 6 من جديد. لهذا السبب يدعى البروتوكول Go-back-N.

- **قياس نافذة الإرسال**

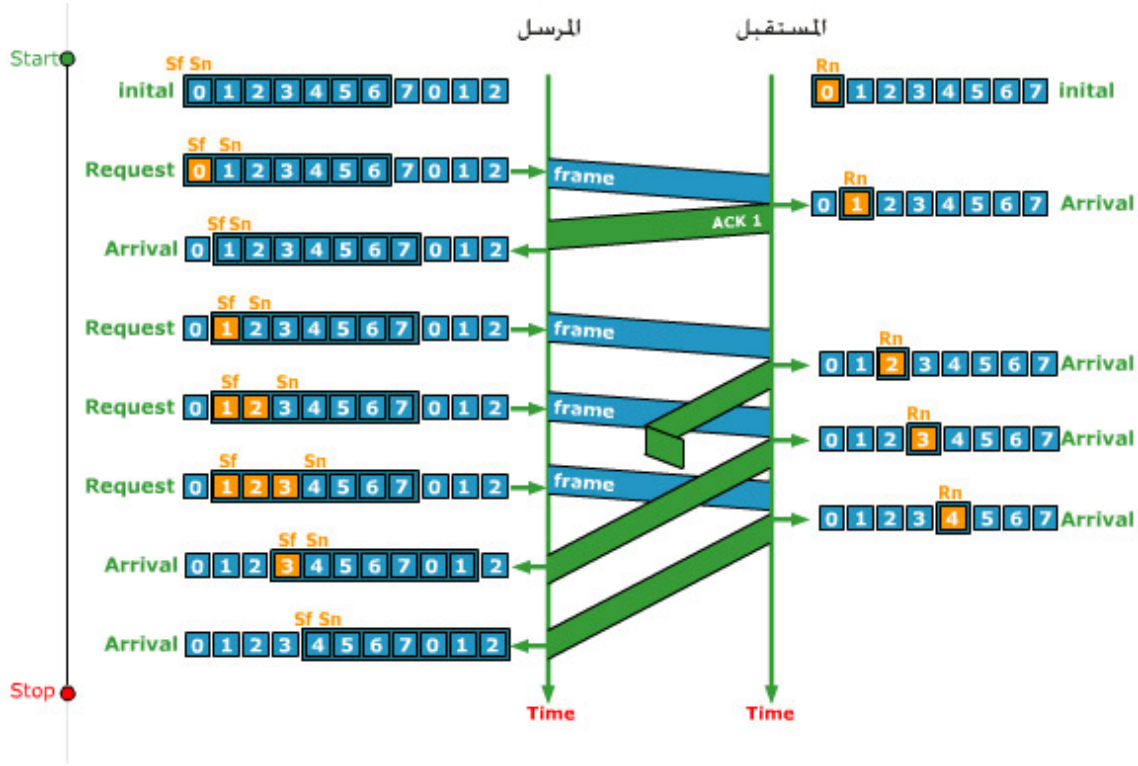
لنناقش الآن سبب اختيار قياس نافذة الإرسال أصغر تماماً من القيمة 2^m. لذلك لنأخذ على سبيل المثال m = 2 مما يعني أن قياس النافذة يجب أن يكون 3. يبين الشكل في الأسفل مقارنة بين نافذة قياسها 3 ونافذة أخرى قياسها 4. فعندما يكون قياس النافذة 3 وتضيع الإقرارات الثلاثة فعندما ينقضي المؤقت الزمني للإطار رقم 0 وتجري إعادة إرسال جميع الأطر الثلاثة بدءاً بالإطار رقم 0 فإن المستقبل الذي ينتظر الآن الإطار رقم 3 سوف يهمل الإطار رقم 0 لأنه تكرار وهو الأمر الصحيح. بينما في حالة قياس النافذة مساوٍ للقيمة 4 وفي حال ضياع جميع الإقرارات فإن المرسل سيعيد إرسال الإطار رقم 0 الذي ينتظره المستقبل.

لذلك فإن المستقبل سيخطئ بقبول هذا الإطار على أساس إنه بدء نافذة جديدة وليس إطاراً مكرراً.



مثال 2

يبين الشكل التالي مثالاً على عمل بروتوكول Go-Back-N ARQ في حالة كون القناة من المرسل إلى المستقبل موثوقة بينما القناة العكسية غير موثوقة. أو، بشكل آخر، لا يجري إضاعة أطر المعلومات وإنما أطر الإقرارات هي التي يمكن أن تضيع أو تتأخر في الوصول.



مخطط التسلسل الزمني للمثال 2

يوجد لدينا، بعد الإقلاع، سبعة أنواع من الأحداث. أحداث الطلب Request يتم قدحها من خلال وجود معطيات قادمة من طبقة الشبكة؛ أحداث الوصول Arrival يتم قدحها من خلال استلام إقرارات من الطبقة الفيزيائية؛ لا يوجد أحداث انقضاء المؤقت الزمني لأننا افترضنا أنه يجري استلام الإقرارات ضمن الوقت؛ لاحظ أنه مع العلم أن الإقرار Ack 2 يضيع فإن Ack 3 سيتلافى الضياع ويفهم على أنه إقرار جميع الأطر السابقة بما فيها الإطار رقم 2.

يوجد أيضاً أربعة أحداث للمستقبل، يجري قدحها جميعاً بوصول إطار من الطبقة الفيزيائية.

3.2.2 بروتوكول تكرار الطلب الآلي مع تكرار انتقائي Selective Repeat Automatic Repeat Request

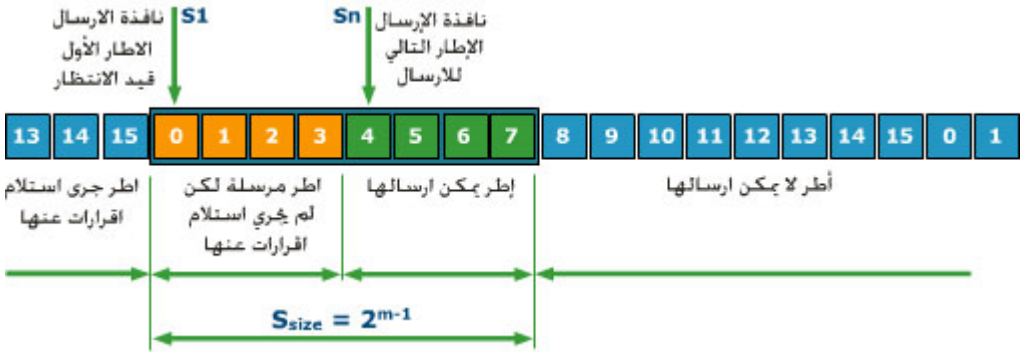
يبسط بروتوكول Go-Back-N ARQ العمل لدى المستقبل لأنه يحتاج لمتابعة متحول واحد فقط دون الحاجة لتخزين الأطر المستقبلة خارج الترتيب. لكن هذا البروتوكول يصبح غير فعالاً عند استخدامه مع وصلات ذات ضجيج كبير Noisy حيث يزداد احتمال تشوه الأطر وإعادة الإرسال الأمر الذي يستهلك حزمة المرور ويبطئ حركة النقل. يوجد لهذه الظروف بروتوكولاً آخر لا يستدعي إعادة نقل عدة أطر وإنما الإطار المشوه فقط؛ يدعى هذا البروتوكول Selective Repeat ARQ. يعتبر هذا البروتوكول أكثر فعالية عندما يوجد ضجيج كبير لكنه يتطلب معالجة أكبر لدى المستقبل.

النافذة

يستخدم بروتوكول التكرار الانتقائي نافذتين أيضاً: نافذة إرسال ونافذة استقبال. يختلف هذا البروتوكول عن البروتوكول السابق بما يلي:

1. يكون قياس نافذة الإرسال أصغر بكثير من البروتوكول السابق وهي 2^{m-1}.
2. يكون قياس نافذة المستقبل مساوٍ لقياس نافذة المرسل.

يستخدم البروتوكول نفس متحولات Go-Back-N ARQ التي ناقشناها سابقاً. يبين الشكل التالي مثالاً عن قياس النافذة المستخدم.



نافذة الإرسال لبروتوكول Selective Repeat ARQ

أما بالنسبة لنافذة المستقبل فهي مختلفة تماماً عن تلك المستخدمة في بروتوكول Go-Back-N ARQ. أولاً، قياس النافذة مساوٍ لنافذة المرسل. ثانياً، يسمح بروتوكول التكرار الانتقائي بقبول أطر خارج الترتيب لكن حسب ما يحدده قياس نافذة المستقبل. فعندما يستطيع المستقبل ترتيب مجموعة من الأطر السابقة فإنه يرسلها إلى طبقة الشبكة.

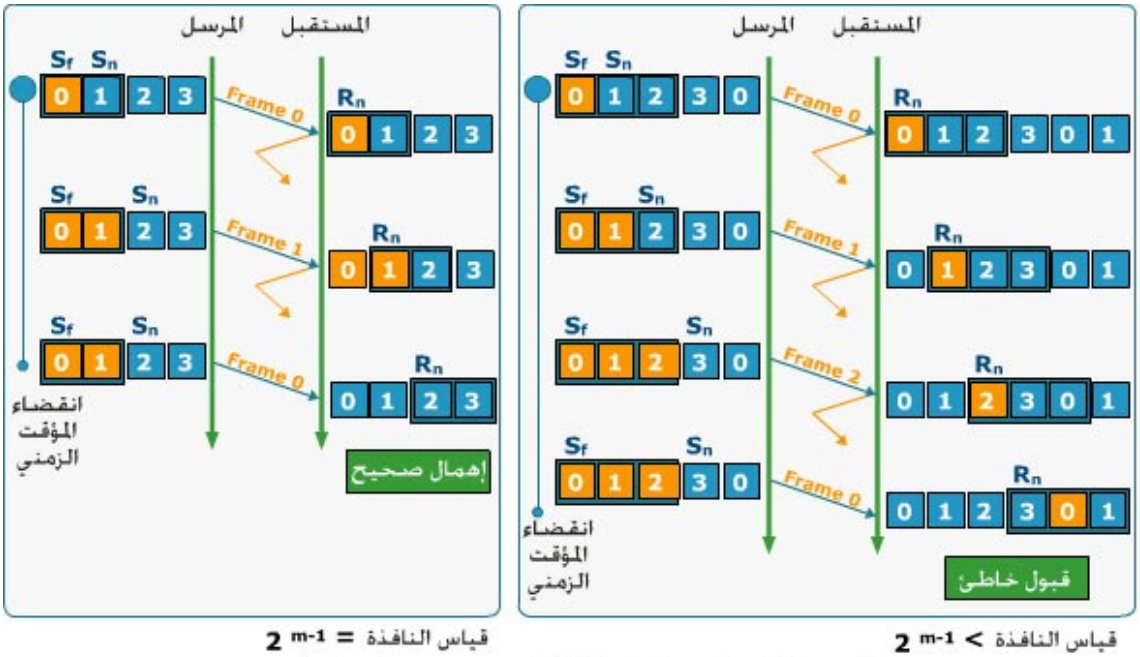
يبين الشكل التالي نافذة المستقبل المستخدمة ضمن هذا البروتوكول.

نافذة المستقبل في بروتوكول Selective Repeat ARQ

لاحظ أن الأطر الحمراء الموجودة داخل نافذة المستقبل تدل على أطر مستقبلة خارج الترتيب والتي تنتظر وصول الأطر المكملة لها حتى يتم تسليمها لطبقة الشبكة.

قياس النافذة

يمكننا الآن مناقشة سبب اختيار قياس النافذة نصف 2^m. لنأخذ مثالاً حيث m=2 مما يعني أن قياس النافذة سيكون 2. يقارن الشكل التالي حالتي قياس النافذة وفق حالتين 2 و 3.



- قياس النافذة لبروتوكول Selective repeat ARQ -

لاحظ أنه إذا كان قياس النافذة هو 2 وضاعت جميع الإقرارات فستنقضي فترة المؤقت الزمني للإطار رقم 0 ويعاد إرساله بينما تتوقع نافذة المستقبل الإطار رقم 2 لذلك يجري إهمال هذا الإطار إهمالاً صحيحاً على أنه إطار مكرر. أما إذا كان قياس نافذة المستقبل هو 3 وتضيع جميع الإقرارات فإن المرسل سيرسل إطاراً مكرراً للإطار رقم 0 بينما نافذة المستقبل تتوقع استقبال الإطار رقم 0 لذلك فإنها ستقبله قبولاً خاطئاً على أنه أول إطار يتبع الدورة الثانية.

```
1  sw = 2m-1;
2  sf = 0;
3  sn = 0;
4
5  While(true)                                  //تكرار إلى اللانهاية
6  {
7    WaitForEvent();                            //انتظار وقوع حدث
8    If(Event(RequestToSend))                   //يوجد طرد للإرسال ويسمح بالإرسال
9    {
10     If (sn – sf >= sw)                       //النافذة ممتلئة
11
12       Sleep();
13
14     GetData();
15
16     MakeFrame(sn);
17
18     StoreFrame(sn);
19
20     SendFrame(sn);
21
22     sn = sn + 1;
23
24     StartTimer(sn);
25
26   }
27
```

```
If(Event(ArrivalNotification))
{
 Receive(Frame);
 If(Corrupted(frame))
  Sleep();
 If(FrameType == NAK)
  If(nakNo between sf and sn)
  {
   Resend(nakNo);
   StartTimer(nakNo);
  }
 If (FrameType == ACK)
  If (ackNo between sf and sn)
  {
   While (sf < ackNo)
   {
```

```
                Purge(Sf);

                StopTimer(sf);

                sf = sf + 1;

            }

          }

    }



    If (Event(TimeOut(t)))

    {

     Start(Timer(t));

     SendFrame(t);

    }

}
```

خوارزمية 6 – خوارزمية المرسل لبروتوكول Selective repeat ARQ

```
1   Rn = 0;
2   NakSent = false;
3   AckNeeded = false;
4   Repeat (for all slots)
5   Marked(slot) = false;
6
7   While(true)
8   {
9     WaitForEvent();
10
11
12
13    If(Event(ArrivalNotification))
14
15    {
16
17     ReceiveFrame();
18
19     If((correptud(frame)) AND (NOT
    Naksent))
20
21     {
22
23       SendNAK(Rn);
24
25       NakSent = true;
26
27       Sleep();
28
29     }
```

```
    If((seqNo <> Rn) AND (NOT NakSent))
    {
     SendNAK(Rn);
      NakSent = true;
     If((seqNo in window) AND (NOT
Marked seqNo))
     {
      StoreFrame(seqNo);
              Marked(seqNo) = true;
              While(Marked(Rn))
              {
   DeliverData(Rn);
   Purge(Rn);
   Rn = Rn +1;
   AckNeeded = true;
      }
              If (AckNeeded)
      {
              SendAck(Rn);
```

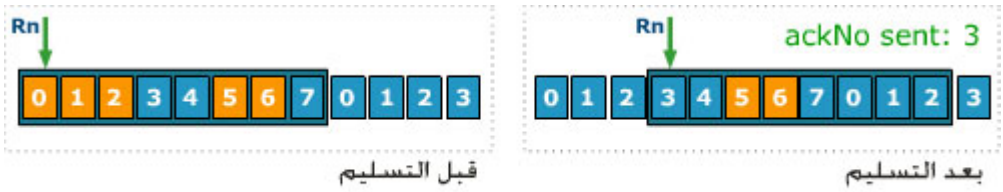
```
7
3                AckNeeded = false;
8
3                NakSent = false;
9
4
0          }
4
1        }
4
2      }
4
3    }
4
4  }
```

خوارزمية 7 – خوارزمية المستقبل لبروتوكول Selective repeat ARQ

تحليل البروتوكول

- **المرسل**: فيما يخص معالجة الأحداث المتعلقة بالطلبات Request فهي مشابهة للبروتوكول السابق ما عدى تخصيص مؤقت زمني لكل إطار مرسل. أما طلبات الوصول Arrival Events فهي أكثر تعقيداً لأنه يمكن وصول إطار ACK أو NAK. إذا وصل إطار NAK سليم فإننا سنعيد إرسال الإطار المقابل، أما إذا وصل إطار ACK سليم فإننا سندخل في حلقة مؤلفة من تفريغ الذاكرة من الإطارات وتوقيف المؤقتات الزمنية المقابلة وزلق النافذة إلى اليسار. بالنسبة لانقضاء المؤقت الزمني فهو أبسط هنا لأننا نحتاج فقط لإعادة إرسال الإطار المقابل له.
- **المستقبل**: نستخدم هنا متحول NakSent حتى لا نغرق المرسل بأطر NAK ومتحول AckNeeded لمعرفة متى يجب إرسال ACK. القيمة البدائية لكليهما تكون False. نستخدم أيضاً مجموعة من المتحولات لتعليم الخانات ضمن نافذة المستقبل عندما يجري استقبال الأطر المقابلة وتخزينها. إذا استقبلنا إطار فاسد ولم نكن قد أرسلنا بعد إطـار NAK فنقوم بإرسال NAK لإعلام الطرف الأخر بأننا لم نستلم الإطار المتوقع. إذا لم يكن الإطار المستلم فاسداً وكان رقمه التسلسلي ضمن النافذة فإننا نخزن الإطار ونعلم الخانة . إذا جرى تعليم

مجموعة متتابعة من الأطر التي تبدأ بالإطار Rn فإننا فيجري تسليم المعطيات المحتواة ضمنها إلى طبقة الشبكة ونقوم بزلق النافذة كما هو موضح بالشكل التالي.

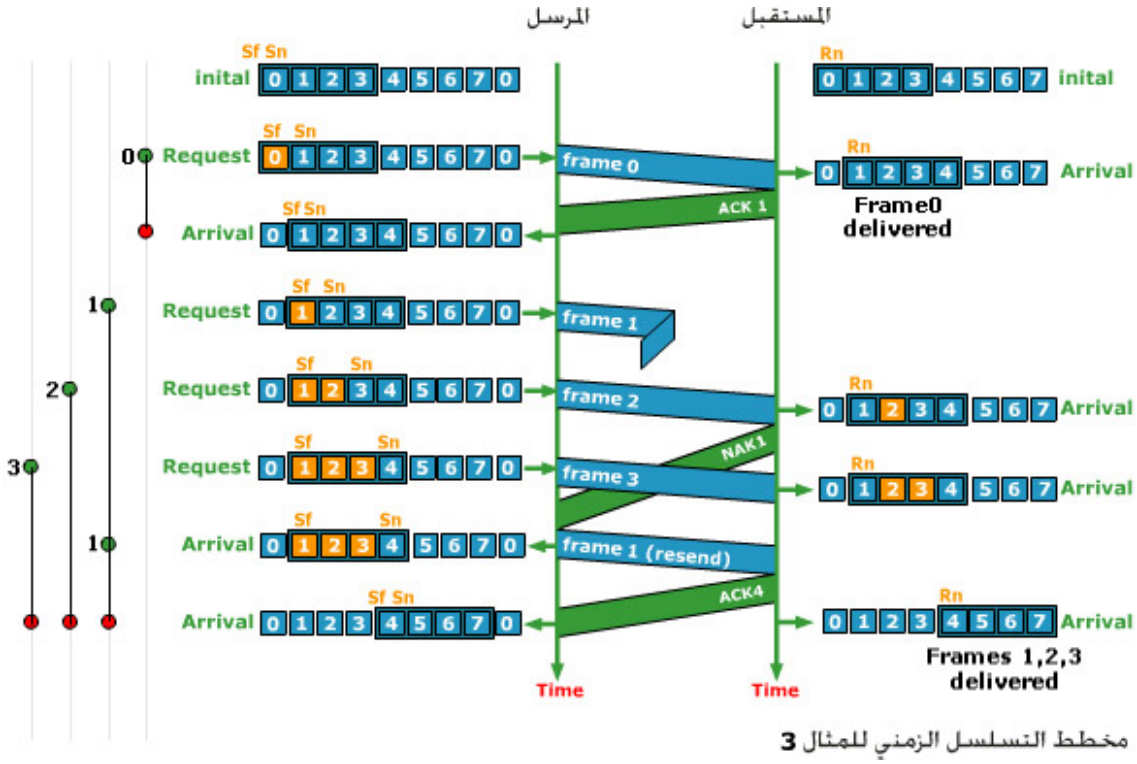


- تسليم المعطيات باستخدام Selective Repeat ARQ -

تحليل البروتوكول(المستقبل)

مثال 3

يبين الشكل مثالاً عن سلوك بروتوكول Selective repeat ARQ عند ضياع الإطار رقم 1.



مخطط التسلسل الزمني للمثال 3

لاحظ أننا نحتاج هنا إلى مؤقت زمني لكل إطار يجري إرساله أو إعادة إرساله لذلك فإننا نحتاج على ترقيم المؤقتات الزمنية حسب رقم الإطار. يقلع المؤقت الزمني للإطار رقم 0 عند وصول الطلب الأول Request ويتوقف عند وصول الإقرار Ack عن هذا الإطار بينما يقلع المؤقت الزمني للإطار 1 عند وصول الطلب الثاني ويعاد إقلاعه من جديد عند وصول NAK ويتوقف عند وصول آخر ACK.

يجب أن نفرق لدى المستقبل بين قبول إطار ما وتسليمه إلى الطبقة الأعلى. لدى الوصول الثاني، الإطار رقم 2 يصل إلى المستقبل حيث يجري تخزينه وتعليمه (خانة ملونة) لكن لا يتم تسليمه لأننا لم نستقبل بعد الإطار رقم 1. لدى الوصول التالي يجري استقبال وتخزين الإطار رقم 3 دون التسليم إلى الطبقة الأعلى أيضاً. فقط لدى الوصول الأخير للإطار رقم 1 يجري تسليم الأطر 1,2,3 إلى الطبقة الأعلى.

يوجد شرطان أساسيان لتسليم الأطر إلى طبقة الشبكة (الأعلى): أولا يجب أن تكون الأطر المستلمة متتابعة؛ وثانياً، يجب أن يبدأ الإطار الأول مع بداية النافذة المنزلقة.

لاحظ أيضاً أنه يجري إرسال إطار NAK بعد الوصول الثاني وليس بعد الوصول الثالث. كما أنه يجري إرسال إطار NAK مرة لكل نافذة ويحدد الخانة الأولى من النافذة.

لاحظ أيضاً أنه جرى إرسال إقرارين، الإقرار الأول يؤكد وصول الإطار الأول بينما الإقرار الثاني فإنه يؤكد وصول ثلاثة أطر. يجري إرسال الإقرارات عند تسليم الأطر إلى طبقة الشبكة أي أنه إذا جرى تسليم عدة أطر دفعة واحدة إلى طبقة الشبكة فإنه يجري إرسال إقرار واحد فقط.

الوحدة الخامسة
الشبكات المحلية LANs

تعريف- الشبكة المحلية Local Area Network (LAN): هي شبكة حواسيب يجري تصميمها لتخديم مسافة جغرافية محدودة كبناء أو حرم.

مع أنه يمكن استخدام الشبكة المحلية داخل مؤسسة ما بشكل معزول لتحقيق المشاركة على الموارد إلا أنه يجري وصل غالبية الشبكات المحلية الحالية إلى الشبكات الواسعة أو إلى الإنترنت.

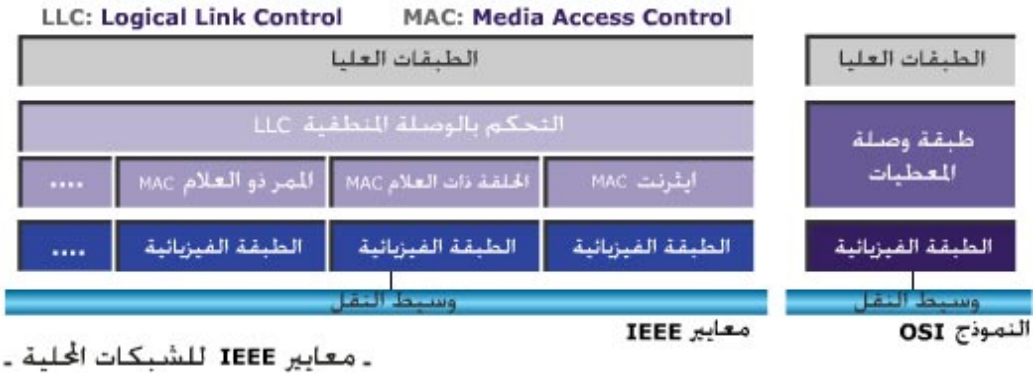
لقد عرفت الأسواق أنواعاً عدة من الشبكات المحلية مثل الإيثرنت Ethernet والحلقة ذات العلام Token Ring والممر ذو العلام Token Bus وشبكات الألياف الضوئية FDDI وشبكات النقل غير المتزامنLAN ATM . بقي بعض هذه الشبكات حياً لفترة من الزمن إلا أن شبكة إيثرنت بقيت هي التقانة المهيمنة حتى وقتنا هذا.

سنخصص هذه الوحدة لشبكة إيثرنت التي شهدت تطوراً ملحوظاً ممثلاً بأربعة أجيال خلال العقد الأخير من الزمن اقتضته الحاجة إلى مواكبة حركة ومتطلبات السوق غير أن المفاهيم الرئيسة لشبكة إيثرنت بقيت نفسها.

1. معايير IEEE

بدأت، خلال عام 1985، منظمة المعيرة IEEE مشروعاً أطلقت عليه اسم Project 802 مهمته وضع المعايير التي تسمح بربط تجهيزات من موردين مختلفين. لم يكن الهدف من هذا المشروع استبدال النموذج المرجعي OSI وإنما كُلِف بتوصيف وظائف الطبقة الفيزيائية وطبقة وصلة المعطيات للبروتوكولات المستخدمة ضمن الشبكات المحلية.

يبين الشكل العلاقة بين النموذج المرجعي OSI ومعايير IEEE 802.



- معايير IEEE للشبكات المحلية -

لاحظ أن معهد IEEE قسم طبقة وصلة المعطيات إلى طبقتين جزئيتين: التحكم بالوصلة المنطقية Logical Link Control (LLC) والتحكم بالولوج إلى الوسيط Media Access Control (MAC).
قام معهد IEEE أيضاً بتعريف طبقات فيزيائية مختلفة لكل نوع من أنواع الشبكات المحلية.

1.1 التحكم بالوصلة المنطقية LLC

تزود هذه الطبقة الجزئية بروتوكولاً وحيداً للتحكم بطبقة وصلة المعطيات من أجل جميع الشبكات المحلية، بعكس التحكم بالولوج إلى الوسيط المتعلق بالشبكة المحلية. يساعد توحيد LLC في تسهيل عملية ربط شبكات محلية مختلفة فيما بينها وذلك لأنه يخفي جميع الاختلافات الموروثة من طبقة MAC.
تتمثل العمليات الأساسية التي تقوم بها طبقة LLC في التأطير والتحكم بالأخطاء وبالتدفق. نحتاج إلى هذا النوع من التحكم ضمن الشبكات المحلية المعزولة لكن عندما نستخدم طبقات عليا مثل TCP/IP فلا تعود هناك حاجة لهذا النوع من التحكم.

2.1 التحكم بالولوج إلى الوسيط MAC

تعرف طبقة MAC الجزئية طرق التحكم بالوسيط لكل نوع من أنواع الشبكات المحلية. فشبكة إيثرنت تستخدم بروتوكول التنصت على الحامل مع الولوج المتعدد وكشف التصادم Carrier Sense Multiple Access with Collision Detection (CSMA/CD) بينما تستخدم الحلقة ذات العلام طريقة أخرى تتمثل بمرور علام على الشبكة والمحطة التي تمتلك العلام تستطيع الإرسال.
تقوم أيضاً هذه الطبقة الجزئية بمعالجة جزء من وظائف التأطير.

2. إيثرنت المعيارية

جرى تعريف شبكة إيثرنت عام 1976 ضمن مخابر شركة Xerox. شهدت إيثرنت، منذ ذلك الحين، أربعة أجيال وهي:

- أيثرنت المعيارية العاملة بسرعة 10 Mbps.
- إيثرنت السريعة العاملة بسرعة 100 Mbps.
- الجيغابت إيثرنت العاملة بسرعة 1 Gbps.
- 10 جيغا إيثرنت العاملة بسرعة 10 Gbps.

1.2 طبقة MAC الجزئية

تقوم هذه الطبقة، في إيثرنت المعيارية، بالتحكم بالولوج إلى الوسيط وتأطير المعطيات المستقبلة من الطبقات العليا تمهيداً لنقلها إلى الطبقة الفيزيائية.

1. بروتوكول CSMA/CD

يطلب هذا البروتوكول من كل محطة التنصت إلى وسيط النقل (اختبار حالة وسيط النقل) قبل أن تقوم بالإرسال فإذا كان يوجد محطة أخرى تستخدم وسيط النقل (أي أنه توجد إشارات على وسيط النقل) فإن المحطة المتنصتة تؤجل عملية الإرسال ريثما تنتهي المحطة المرسلة من وسيط النقل.

في حال وجدت محطتان تتنصتان على وسيط النقل في نفس اللحظة فإن تلك المحطتين سوف ترسلان في نفس الوقت وتتصادم الإشارتان مع بعضهما البعض وهذا ما يسمى بالتصادم Collision الذي يمكن أن يحدث أيضاً في حال بدأت إحدى المحطات بالإرسال لكن أول بت مرسل لم يصل بعد إلى المحطة المتنصتة (لاحظ أن سرعة انتشار الإشارات هي من مرتبة سرعة الضوء أي 3 x 10^8 m/s) التي سوف ترسل أيضاً ويحدث التصادم. لذلك يطلب بروتوكول CSMA/CD من كل محطة مراقبة وسيط النقل أثناء فترة الإرسال للتأكد من نجاح الإرسال. في حال عدم نجاح الإرسال فإنه يجب على المحطة إعادة الإرسال بعد انتظار زمن عشوائي.

- الطول الأصغري لإطار إيثرنت: حتى يعمل بروتوكول CSMA/CD بشكل جيد فإننا نحتاج إلى تحديد الطول الأصغري للإطار وذلك لأنه يجب على أي محطة أن تكتشف التصادم قبل أن تنتهي من إرسال آخر بت فتتوقف عن الإرسال وذلك لأنها لا تتنصت على الحامل بعد انتهاء الإرسال. أو بشكل آخر، إذا نجحت أي محطة من إرسال آخر بت دون أن تتحسس حدوث أي تصادم فإنها تعتبر أن الإرسال قد نجح فتحذف الإطار من الذاكرة وتنتقل إلى الإطار التالي.

  بناء على ما سبق، فإن زمن إرسال الإطار T_{fr} يجب أن يكون على الأقل ضعف زمن الانتشار الأعظمي T_p. لنفترض أن المحطتين المتورطتين في التصادم تقعان في طرفي الكابل (أكبر مسافة ممكنة بينهما) لذلك يلزم الإشارة لتصل من المحطة الأولى إلى الثانية زمناً قدره Tp وعندما يقع التصادم فإنه يلزمنا أيضاً Tp ثانية لتصل إشارة التصادم إلى

المحطة الأولى؛ أي أنه يجب على المحطة الأولى أن تكون قيد الإرسال بعد مضي 2 x Tp ثانية.

مثال 1

لتكن لدينا شبكة تستخدم بروتوكول CSMA/CD مزودة بعرض حزمة قدره 10 Mbps. فإذا اعتبرنا أن الزمن الأعظمي للانتشار (شاملاً جميع أنواع التأخير ضمن الأجهزة المختلفة) هو 25.6 µs؛ احسب الطول الأصغري للإطار.

الحل:

زمن نقل الإطار هو:

$$T_{fr} = 2 \times T_p = 51.2\ \mu s$$

هذا يعني أنه في أسوء الحالات فإن أي محطة تحتاج لأن ترسل لفترة 51.2 µs على الأقل حتى تستطيع اكتشاف التصادم.

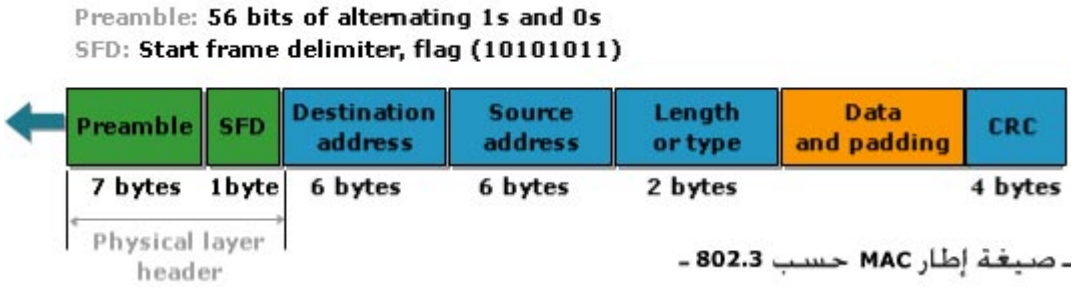
الطول الأصغري للإطار:

$$10 \times 10^6 \times 51.2 \times 10^{-6} = 512 \text{ bits (64 bytes)}$$

هذا هو الطول الأصغري الحقيقي لإطار إيثرنت التقليدية.

- صيغة الإطار Frame Format

يبين الشكل التالي صيغة إطار إيثرنت.



- صيغة إطار MAC حسب 802.3 -

لا تؤمن إيثرنت أية آلية لإقرار استلام الأطر الأمر الذي يجعلها غير موثوقة إلا أنه يمكن الاعتماد على الطبقات العليا لتحقيق الوثوقية المطلوبة.

- **الممهد Preamble**: هو عبارة عن سبعة بايتات مؤلفة من واحدات وأصفار متعاقبة تستخدم لتحذير المحطة المستقبلة إلى الإطار القادم وتمكنها من مزامنة مؤقتها الزمني مع مؤقت المحطة المرسلة.
- **محرف تحديد بداية الإطار Start Frame Delimiter (SFD):** وهو عبارة عن بايت واحد (10101011) يدل على بداية الإطار. عندما لا يكون المستقبل متزامناً مع المرسل

فإنه من الممكن أن يضيع بعض البتات فتصبح الطريقة الوحيدة لاكتشاف بداية الإطار هي بوجود واحدين متعاقبين.

- **عنوان الوجهة Destination Address (DA):** وهو عبارة عن 6 بايتات يحوي على العنوان الفيزيائي للمحطة الوجهة أو المحطات الموجه إليها الإطار.
- **عنوان المصدر Source Address (SA):** وهو يتألف أيضاً من 6 بايتات ويحوي على العنوان الفيزيائي للمرسل.
- **الطول أو النوع Type or Length:** استخدمت إيثرنت الأصلية هذا الحقل للدلالة على نوع بروتوكول الطبقة العليا الذي يستخدم الإطار. أما معايير IEEE فتستخدم هذا الحقل للدلالة على الطول أو عدد البايتات الموجودة ضمن حقل المعطيات.
- **المعطيات Data:** يحوي هذا الحقل المعطيات المغلفة من بروتوكول الطبقة العليا. يجب أن تكون المعطيات ضمن المجال (46 bytes – 1500 bytes).
- **اختبار التكاملية CRC:** معلومات خاصة باكتشاف الأخطاء حيث يتم استخدام CRC-32.

- طول الإطار:

يعود سبب تحديد الطول الأصغري لإطار إيثرنت إلى حاجة بروتوكول CSMA/CD الذي يتطلب إطاراً ذو طول 64 بايت مع الترويسة واللاحقة. وبما أن الترويسة واللاحقة، دون حقل Preamble and SFD، يشغلان 18 بايتاً فيصبح الطول الأصغري للمعطيات هو 46 بايت. فإذا كانت المعطيات القادمة من الطبقة الأعلى أقل من 46 بايت فيجب إضافة حشوة Padding لملئ الفراغ.

أضف إلى ذلك أن المعايير تحدد الطول الأعظمي للإطار، بدون Preamble and SFD ، بـ ـ bytes 1518 فإذا حذفنا منها 18 بايت فيبقى لدينا bytes 1500 تشكل الطول الأعظمي للمعطيات القادمة من الطبقة العليا.

2. العنونة Addressing

تحوي كل محطة مربوطة إلى شبكة إيثرنت بطاقة شبكة Network Interface Card (NIC) . تمتلك كل بطاقة شبكة عنوان فيزيائي مكون من 6 بايتات تكتب كالمثال التالي:

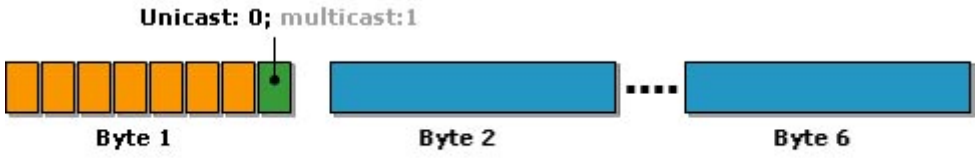
06 : 01 : 02 : 01 : 2C : 4B

لاحظ أن كل جزء من العنوان السابق يمثل بايتاً واحداً مكتوباً بالترميز الست عشري.

3. العناوين وحيدة الوجهة ومتعددة الوجهات والمعممة

يكون عنوان المصدر دائماً وحيد الوجهة Unicast وذلك لأن الإطار يأتي دائماً من محطة واحدة فقط. أما بالنسبة لعنوان الوجهة فيمكن أن يكون وحيد الوجهة أو متعدد الوجهـات Multicast أو معمم Broadcast.

يمكن التمييز بين هذه الأنواع المختلفة عن طريق البت الأقل دلالة Least Significant bit من البايت الأول (إلى اليسار) لعنوان الوجهة: فإذا كانت قيمة البت 0 فالعنوان وحيد الوجهة وإذا كانت القيمة 1 فالعنوان متعدد الوجهات. كما هو موضح في الشكل التالي:



ـ العناوين وحيدة الوجهة و متعددة الوجهات ـ

يعرف العنوان وحيد الوجهة مستقبل وحيد أي أن العلاقة بين المرسل والمستقبل هي علاقة واحد إلى واحد. أما العنوان متعدد الوجهات فهو يعرف مجموعة من المستقبلين وتصبح العلاقة بـين المرسل والمستقبل هي علاقة واحد إلى مجموعة.

بالنسبة لعنوان التعميم (أو العنوان المعمم) فهو حالة خاصة من العنوان متعدد الوجهات حيث يكون المستقبلون هم جميع المحطات الموجودة على الشبكة وتكون جميع بتات العنوان مساوية 1 أي All ones.

تمرين

عرف نوع عناوين الوجهات التالية:

4A : 30 : 10 : 21 : 10 : 1A
47 : 20 : 1B : 2E : 08 : EE
FF : FF : FF : FF : FF : FF

يجب الانتباه هنا إلى أن إرسال البايتات يجري من اليسار إلى اليمين (بايتاً بايتاً) وبالنسبة لكل بت داخل البايت فيجري من اليمين إلى اليسار (البت الأقل دلالة أولاً). لذلك فإن البايت A4 يرسل أولاً في الحالة الأولى كما يجري إرسال بت تحديد نوع العنوان ضمن هذا البايت أولاً.

مثال

بين تسلسل إرسال العنوان:

47 : 20 : 1B : 2E : 08 : EE

على وسيط النقل.

الحل

01110111 00010000 01110100 11011000 00000100 11100100 ←

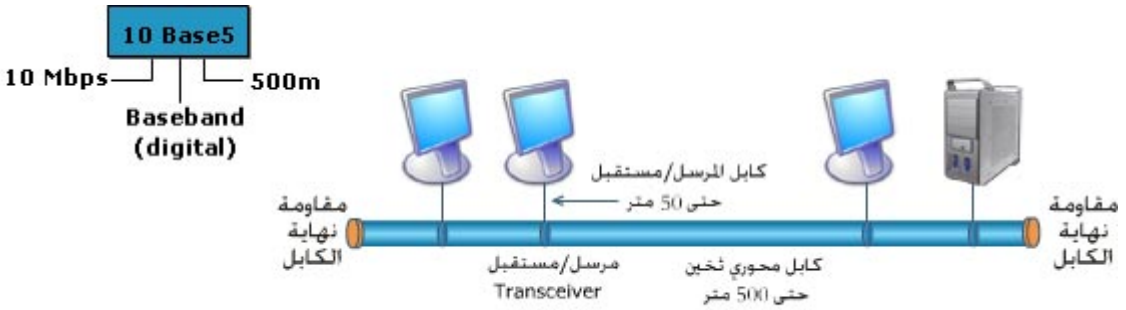
4. الطبقة الفيزيائية

تعرف معايير إيثرنت أنواعاً عدة للطبقة الفيزيائية، سنناقش أهم أربعة منها وهي:

10Base5
10Base2
10Base-T
10Base-F

- **إيثرنت الثخينة 10 Base 5**

تعود التسمية إلى ثخانة الكابل المستخدم لتحقيقها. لقد كانت شبكة إيثرنت الثخينة أول شبكة إيثرنت تستخدم طبولوجية ممرية مع مرسل/مستقبل Transceiver خارجي يتم ربطه عن طريق فرعة Tap إلى كابل محوري ثخين. يبين الشكل التالي مخططاً لهذا النوع من الشبكات.



- تحقيق ايثرنت الثخينة 10 Base5 -

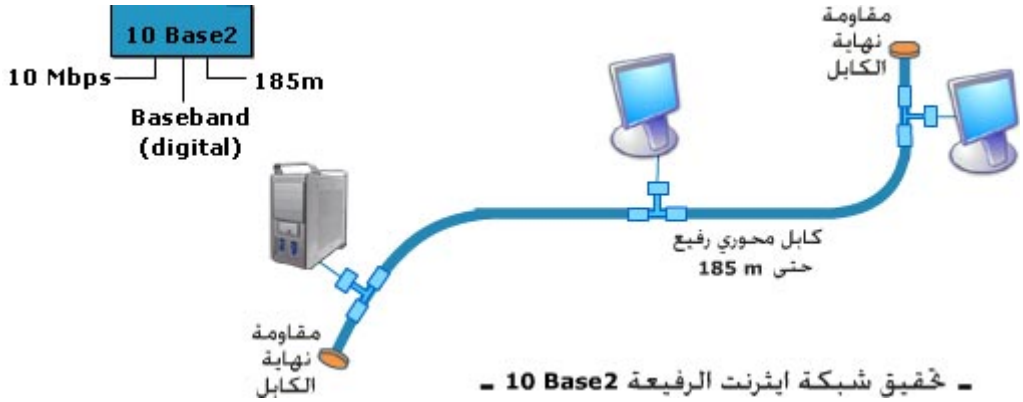
تتمثل مسؤولية المرسل/مستقبل في النقل والاستقبال واكتشاف التصادم. يجري وصل المرسل/مستقبل على محطة عن طريق كابل خاص بالمرسل/مستقبل يؤمن مسارين منفصلين للإرسال وللاستقبال، مما يعني أن التصادمات يمكن أن تقع على الكابل المحوري فقط.

يجب أن لا يزيد طول الكابل المحوري عن 500 متر وإلا فإن الإشارات ستتشوه نتيجةً للتخامد. غير أنه من الممكن استخدام 4 مكررات (أو 5 مقاطع شبكة) على الأكثر مما يوسع قطر الشبكة إلى 2500 متر.

- **إيثرنت الثخينة 10 Base 2**

تستخدم إيثرنت الرفيعة أيضاً الطبولوجية الممرية لكن الكابلات المحورية تكون أقل ثخانةً وأكثر مرونةً لذلك يمكننا ثنيها بحيث تستطيع المرور بالقرب من المحطات. يصبح، في هذه الحالة،

المرسل/مستقبل جزءاً من بطاقة الشبكة الموضوعة ضمن المحطة. يبين الشكل التالي مخططاً لهذا المعيار.

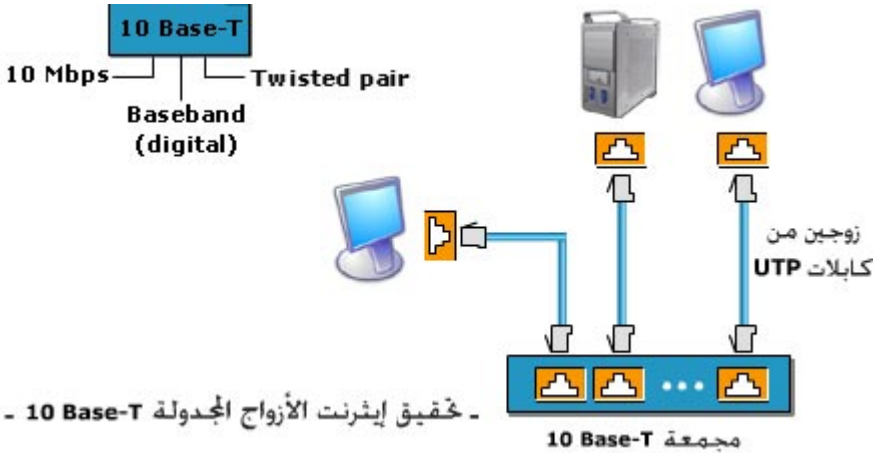


ـ تحقيق شبكة ايثرنت الرفيعة 10 Base2 ـ

يعتبر تحقيق هذا النوع من الشبكات أقل كلفةً من النوع السابق وتمديده أسهل. لكن طول الكابل ينخفض إلى 185 متر بسبب مستوى التخامد الكبير الذي يعاني منه.

- **إيثرنت الأزواج المحدولة 10Base-T**

يستخدم هذا النوع من شبكات إيثرنت طبولوجية نجمية حيث يجري ربط جميع المحطات إلى مجمعة مركزية Hub باستخدام زوجين مجدولين كما هو موضح في الشكل التالي.



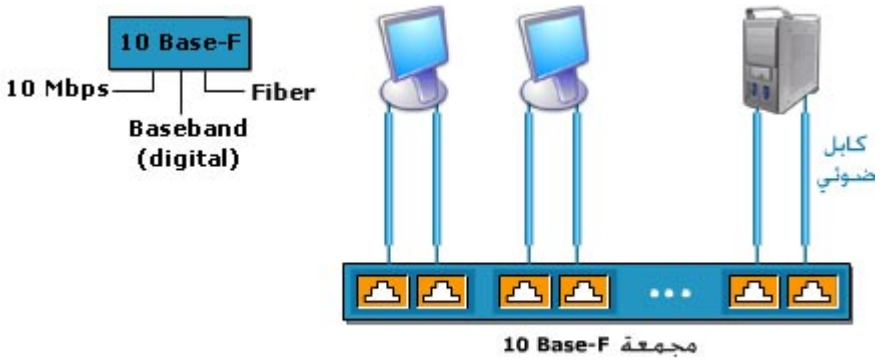
ـ تحقيق إيثرنت الأزواج المجدولة 10 Base-T ـ

لاحظ أن الزوجين المجدولين يشكلان مسارين (الأول للإرسال والثاني للاستقبال) بين المحطة والمجمعة؛ لذلك فإن التصادمات تحدث ضمن المجمعة.

يحدد طول الأزواج المجدولة بين أي محطة وبين المجمعة بـ 100 متر لتقليل أثر التخامد قدر الإمكان.

- **إيثرنت الضوئية 10 Base-F**

تستخدم شبكة إيثرنت الضوئية طبولوجية نجمية لربط المحطات إلى المجمعة، عن طريق كابل ضوئي مؤلف من ليفين كما هو موضح بالشكل التالي:



- تحقيق إيثرنت الضوئية 10 Base-F -

2.2 التغيرات التي طرأت على معايير إيثرنت

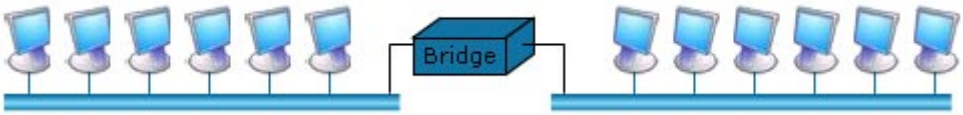
1.2.2 استخدام الجسور

تمثلت الخطوة الأولى من تطور شبكة إيثرنت في تقسيم الشبكة عن طريق الجسور Bridges . يساعد الجسر في رفع عرض الحزمة المتاح وفصل مجالات التصادم.

- **رفع عرض الحزمة**

تجري، ضمن شبكة إيثرنت العادية (أي بدون جسور)، المشاركة بين المحطات على وسيط النقل. فإذا فرضنا أنه لدينا 12 محطة موزعة على وسيط نقل مشترك فإن حصة كل محطة ستكون 12/10 Mbps نظرياً. في حال استخدمنا جسر لتقسيم الشبكة فيصبح لدينا 6 محطات ضمن كل مقطع شبكة فتصبح حصة كل محطة 6/10 Mbps أي أننا ضاعفنا تقريباً حصة كل محطة.

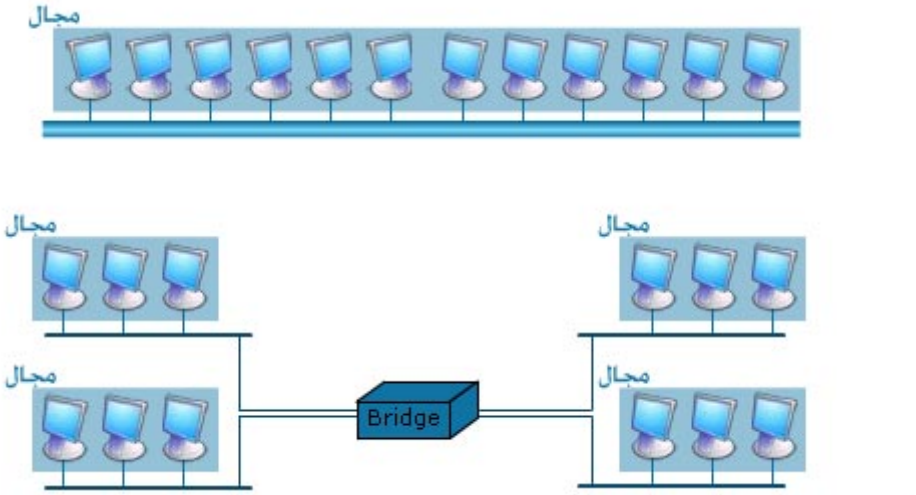
يجب هنا، لتحسين فعالية الشبكة، أخذ المعطيات التي يجري تراسلها بين المقطعين بعين الاعتبار أو بشكل آخر يجب أن لا تزيد كمية المعطيات المتراسلة بين المقطعين عن 20% من عرض الحزمة المتاح.



شبكة مع أو بدون جسر

- **فصل مجالات التصادم**

تتجلى الميزة التالية للجسر في فصل مجالات التصادم. يظهر الشكل التالي مجال التصادم لشبكة بدون ومع جسر.



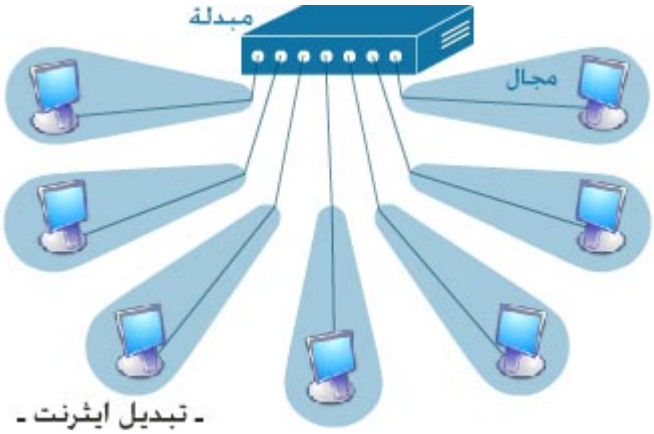
مجال التصادم ضمن شبكة مع أو بدون جسر

لاحظ أن مجال التصادم يصبح أصغر في حالة الجسر أي أن احتمال حدوث تصادم يصبح أقل. فبدون الجسر، يوجد هناك 12 محطة (أي قياس مجال التصادم هو 12) متنافسة للولوج إلى وسيط النقل بينما مع الجسر أصبح عدد المحطات المتنافسة 3 محطات.

2.2.2 إيثرنت المبدلة Switched Ethernet

سنحاول الآن توسيع فكرة الجسر أبعد من ذلك. فبدلاً من وجود شبكتين حتى أربعة، لماذا لا نضع N شبكة حيث يمثل العدد N عدد المحطات (أو الشبكات) داخل الشبكة المحلية؟ سنحصل في هذه الحالة على مبدلة Switch مؤلفة من N بوابة وستجري المشاركة بين المبدلة وكل محطة (أي ستكون حصة المحطة 2/10 = 5 Mbps). أضف إلى ذلك أن عدد مجالات التصادم قد أصبح N.

تعريف: مبدلة من المستوى الثاني Layer-2 Switch: هي جسر مؤلف من N بوابة مضاف إليه بعض التحسينات المتعلقة بسرعة معالجة الأطر. يبين الشكل التالي مخطط لمبدلة.



- **إيثرنت مزدوجة الاتجاه Full-duplex Ethernet**

لعل أحد مساوئ شبكات 10 Base5 و10 Base2هو كونهم وحيدي الاتجاه؛ يمكن لمحطة ما أن ترسل أو تستقبل في لحظة ما ولكن ليس كليهما. لذلك، سعى التطور الأول إلى جعل إيثرنت المبدلة مزدوجة الاتجاه مما يرفع استطاعة كل مجال تصادم إلى 20 Mbps بدلاً عن 10 Mbps. يجري تحقيق ذلك باستخدام وصلتين بين المحطة والمبدلة: الأولى للإرسال والثانية للاستقبال. لاحظ أنه عند استخدام إيثرنت المبدلة مزدوجة الاتجاه، لا تعود هناك حاجة لاستخدام بروتوكول CSMA/CD وذلك لعدم إمكانية حدوث أي تصادم، فكل وصلة بين المبدلة والمحطة هي مجال تصادم قائم بذاته وثنائي الاتجاه.

3. إيثرنت السريعة Fast Ethernet

يوصف المعيار IEEE 802.3u خصائص شبكة إيثرنت السريعة. تتميز شبكة إيثرنت السريعة بكونها متوافقة مع إيثرنت التقليدية لكن بمعدل نقل معطيات أكبر بعشر مرات. يمكننا تلخيص أهداف شبكة إيثرنت السريعة على النحو التالي:

- زيادة معدل نقل المعطيات إلى 100 Mbps
- جعلها متوافقة مع إيثرنت المعيارية
- المحافظة على نفس العناوين الفيزيائية المكونة من 48 bits
- المحافظة على صيغة الإطار نفسها
- المحافظة على نفس الأطوال الصغرى والعظمى للإطار.

1.3 طبقة MAC الجزئية

جرى أثناء تصميم طبقة MAC الجزئية لإيثرنت السريعة استبعاد الطبولوجيات الممرية والاحتفاظ فقط بالطبولوجية النجمية. يبقى لدينا خياران بالنسبة للطبولوجية النجمية: نصف المزدوج والمزدوج. يجري في الحل نصف المزدوج وصل المحطات إلى مجمعة مركزية بينما في الحل المزدوج فيجري وصل المحطات إلى مبدلة مزودة بدارئة Buffer عند كل بوابة. يبقى بروتوكول التحكم بالولوج CSMA/CD نفسه في الحل نصف المزدوج بينما يمكن توقيفه في الحل المزدوج.

- **التفاوض الذاتي Auto-negotiation**

يسمح التفاوض الذاتي لجهازين بالتفاوض على نمط أو معدل نقل المعطيات. لقد جرى تصميمه بشكل خاص لتحقيق ما يلي:

- السماح بوصل أجهزة غير متوافقة مع بعضها البعض. أي يمكن وصل جهاز يعمل بسرعة 10 Mbps مع جهاز آخر يعمل بسرعة 100 Mbps (عن طريق تخفيض سرعته إلى 10 Mbps)
- السماح لجهاز واحد بامتلاك معدلات نقل معطيات مختلفة
- السماح لمحطة ما باختبار مقدرات المجمعة

2.3 الطبقة الفيزيائية

الطبولوجية

يمكن استخدام شبكة إيثرنت السريعة لربط محطتين أو أكثر؛ يجري ربط محطتين فقط عن طريق كابل مباشر ومعكوس بين المحطتين. في حال وجود أكثر من محطتين فإننا نحتاج إلى مجمعة أو مبدلة لتحقيق الربط النجمي.

التحقيق

يمكن تحقيق إيثرنت السريعة على المستوى الفيزيائي باستخدام سلكين أو أربعة أسلاك. يجري استخدام السلكين مع الأزواج المجدولة غير المدعمة Unshielded Twisted Pairs (UTP) Category 5 المستخدمة في Base-TX100 أو الكابلات الضوئية (Base-FX100). بينما يجري استخدام أربعة أسلاك مع كابلات UTP Category 3 فقط (Base-T4100).

4. الجيغابت إيثرنت Gigabit Ethernet

تعمل شبكة Gigabit Ethernet، التي عرفت باسم IEEE 802.3z، بمعدل نقل معطيات يصل إلى 1000 Mbps أو 1 Gbps. جرى طبعاً المحافظة على جميع الأهداف التصميمية التي ذكرناها ضمن إيثرنت السريعة مع إضافة التوافق مع إيثرنت السريعة ودعم التفاوض الذاتي.

1.4 طبقة MAC الجزئية

تدعم الجيغابت إيثرنت نمطي الإرسال أحادي وثنائي الاتجاه. مع أن غالبية شبكات إيثرنت العاملة بسرعة 1 Gbps تتبع نمط الإرسال ثنائي الاتجاه إلا أننا سنناقش نمط الإرسال أحادي الاتجاه المعمول به من أجل المحافظة على التوافق مع الأجيال السابقة.

نمط

- **الإرسال المزدوج**

نستخدم في هذا النمط مبدلة إيثرنت مربوطة إلى جميع المحطات أو بقية المبدلات. تمتلك كل مبدلة دارئة لكل بوابة دخل تخزن فيها الأطر لحين إرسالها. لا يوجد تصادمات ضمن هذا النوع من الإرسال ولا يتم استخدام بروتوكول CSMA/CD. بالنسبة للقيمة العظمى لطول الكابل فيحدده تخامد الإشارة ضمن الكابل وليس بروتوكول التحكم بالولوج.

- **نمط الإرسال نصف المزدوج**

يجري هنا استخدام مجمعة مركزية تقلد عمل الكابل حيث يمكن حدوث التصادمات، لذلك يجب استخدام بروتوكول CSMA/CD. يتعلق الطول الأعظمي للشبكة، في هذه الحالة، بالطول الأصغري للإطار. لذلك فقد جرى تعريف ثلاثة أنماط عمل: تقليدي وتوسيع الحامل ورشق الأطر.

- **نمط العمل التقليدي:**

نحافظ في هذا النمط على طول الإطار الأصغري كما هو عليه الحال في إيثرنت التقليدية (أي 512 Bits). لكن في هذا النمط يصبح قطر الشبكة مساوٍ لـ 25 متر لأننا نرسل بسرعة أكبر بمائة مرة من إيثرنت التقليدية فعدد البتات الموجود على الشبكة يصبح أكبر بمائة مرة فيجب تخفيض قطر الشبكة مائة مرة (من 2500 متر حتى 25 متر). يمكن قبول هذا النمط إذا كانت جميع التجهيزات موجودة ضمن غرفة واحدة لكنه يبقى غير مناسب في الحالة العامة.

- **نمط توسيع الحامل Carrier Extension:**

يجري هنا زيادة طول الإطار الأصغري ليصبح 512 bytes بدلاً عن 64 Bytes. أي أننا ضاعفنا القيمة ثمان مرات. يجبر هذا النمط كل محطة على إضافة حشوة Padding إلى أي إطار أصغر من 512 Bytes. بهذه الطريقة يمكننا مضاعفة الطول الأصغري للشبكة ثمان مرات ليصبح 200 m مع الاحتفاظ بمسافة 100 m بين المجمعة والمحطة (علل لماذا ؟)

رشق الأطر Frame Bursting: جرى طرح هذا النمط لمعالجة عدم فعالية النمط السابق خاصةً في حالة الأطر الصغيرة. فبدلاً عن توسيع الطول الأصغري للإطار يمكننا إرسال عدة أطر معاً (أي ضمن إطار واحد) بعد إضافة حشوات بين الأطر حتى لا تصبح القناة خاملة. أو بشكل آخر، خداع المحطات الأخرى بجعلها تعتقد أننا قيد إرسال إطار كبير وليس عدة أطر.

تبقى الطبولوجية الفيزيائية كما كانت عليه في إيثرنت السريعة. أما بالنسبة لتحقيق الجيغابت إيثرنت فيمكننا تصنيفه أيضاً عن طريق استخدام سلكين أو أربعة أسلاك.

التقانات التي تحقق الجيغابت إيثرنت باستخدام سلكين فقط:

1000Base-SX (Short-wave) Fiber-optic cable
1000Base-LX (Long-wave) Fiber-optic cable
1000Base-CX (Shielded Twisted Pair) STP cable

التقانة التي تستخدم أربعة أسلاك هي تقانة الأزواج المجدولة غير المدعمة Cat. 5e UTP أو أعلى Base-T1000.

5. عشرة جيغابت إيثرنت Ten-Gigabit Ethernet

تعرف العشرة جيغابت إيثرنت باسم IEEE 802.3ae. لقد جرى تصميمها بشكل خاص لتحقيق ما يلي:

- رفع معدل نقل المعطيات إلى 10 Gbps.
- جعلها متوافقة مع إيثرنت التقليدية والسريعة والجيغابت إيثرنت.
- المحافظة على نفس طول العناوين الفيزيائية وهي 48 bits.
- المحافظة على نفس صيغة الإطار.
- المحافظة على القيمة الدنيا والعليا لأطوال الإطارات

- السماح بتحقيق ترابط الشبكات المحلية الموجودة باستخدام شبكة إقليمية MAN أو واسعة WAN
- جعل إيثرنت متوافقة مع تقانات مثل مرحل الأطر Frame Relay والنقل غير المتزامن Asynchronous Transfer mode (ATM).

تعمل شبكة العشرة جيغابت إيثرنت بنمط النقل المزدوج فقط مع استبعاد بروتوكول CSMA/CD.

يبين الجدول التالي كيفية تحقيق شبكة العشرة جيغابت إيثرنت باستخدام الكابلات الضوئية.

المواصفات	10GBase-S	10GBase-L	10GBase-E
طول الموجة	الأمواج القصيرة 850 nm	الأمواج الطويلة 1310 nm	الأمواج الموسعة 1550 nm
وسيط النقل	الألياف للضوئية متعددة الأنماط Multi-mode	الألياف للضوئية وحيدة النمط Single-mode	الألياف للضوئية وحيدة النمط Single-mode
الطول الأعظمي	300 متر	10 كم	40 كم

الوحدة السادسة

ترابط الشبكات Network Interconnections

لا توجد الشبكات المحلية بشكل معزول وإنما يتم ربطها إلى بعضها البعض عن طريق الشبكات الواسعة أو الإنترنت. حتى نستطيع ربط شبكة محلية مع شبكة أخرى فإننا بحاجة إلى أجهزة ربط. يمكن لأجهزة الربط أن تعمل على مستويات متعددة حسب النموذج المرجعي OSI.
سنميز فيما يلي بين أجهزة ربط الشبكات Connecting Devices وأجهزة ترابط الشبكات Interconnecting Devices.

1. أجهزة ربط الشبكات Network Connection Devices

تقسم أجهزة ربط الشبكات إلى خمسة أصناف حسب المستوى الذي تعمل به. يبين الشكل التالي نوع كل جهاز ومستوى عمله.

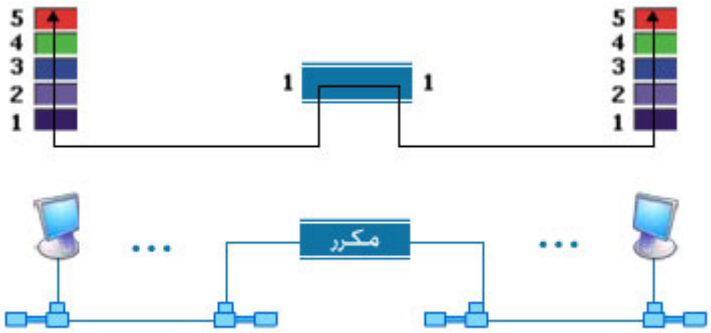
- تصنيف أجهزة ربط الشبكات -

1.1 المجمعة غير النشطة Passive hub

يعمل هذا الجهاز تحت مستوى الطبقة الفيزيائية وهو مجرد مربط Connector يؤمن عملية ربط الأسلاك القادمة من أطراف متعددة. فإذا أخذنا على سبيل المثال شبكة إيثرنت ذات الطبولوجية النجمية، تكون المجمعة غير النشطة عبارة عن نقطة تصادم الإشارات القادمة من مختلف المحطات. يعتبر هذا النوع من المجمعات جزءاً لا يتجزأ من وسيط النقل.

2.1 المكررات Repeaters

يعمل المكرر على المستوى الفيزيائي فقط. يقوم المكرر باستقبال الإشارات التي تحمل المعطيات وإعادة توليدها. تتعرض الإشارات أثناء انتقالها إلى التشوه أو التخامد، لذلك يفيد المكرر بزيادة الطول الفيزيائي لوسيط النقل أو الشبكة المحلية كما هو مبين في الشكل التالي.



مثال عن مكرر يربط مقطعي شبكة محلية

تعريف: نسمي مقطع Segment شبكة محلية الجزء من الشبكة المحصور بين مكررين أو مكرر ومقاومة. لاحظ أن المكرر لا يربط شبكتين محليتين وإنما يربط مقطعي شبكة محلية مع بعضهم البعض لتشكيل شبكة واحدة.

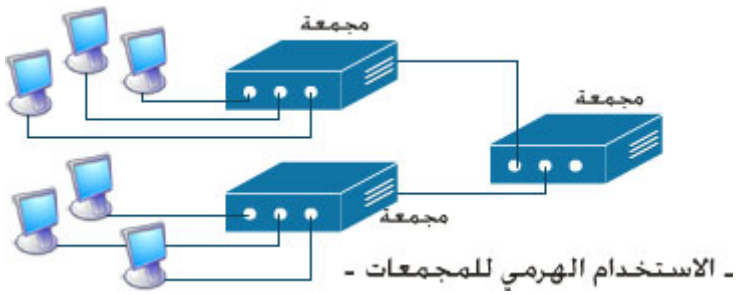
يجري استخدام المكرر في شبكة 5 Base 10 لتجاوز حد 500 متر حيث يمكننا استخدام 4 مكررات على التوالي فتصل المسافة العظمى للشبكة إلى 2500 متر ونسمي الجزء من الشبكة محصور بين مكررين بالمقطع. يجب الانتباه هنا إلى أن مجموعة المقاطع والمكررات تعتبر شبكة محلية واحدة وليست ترابط شبكات محلية.

تتمثل وظيفة المكرر باستقبال الإشارات على بوابة ما وإعادة توليدها ومن ثم إرسالها على البوابة الأخرى بدون أي إمكانية تصفية.

يجب هنا التمييز بين المكرر والمضخم Amplifier؛ فالمضخم يقوم بتضخيم الإشارة والضجيج معاً بينما المكرر يعيد توليد الإشارة الأصلية كما كانت.

3.1 المجمعة النشطة Active Hub

المجمعة النشطة هي مكرر متعدد البوابات. يجري استخدامها عادةً لتحقيق الوصل بين المحطات ضمن طبولوجية نجمية. يمكننا أيضاً استخدام عدة مستويات من المجمعات كما هو موضح في الشكل التالي.



- الاستخدام الهرمي للمجمعات -

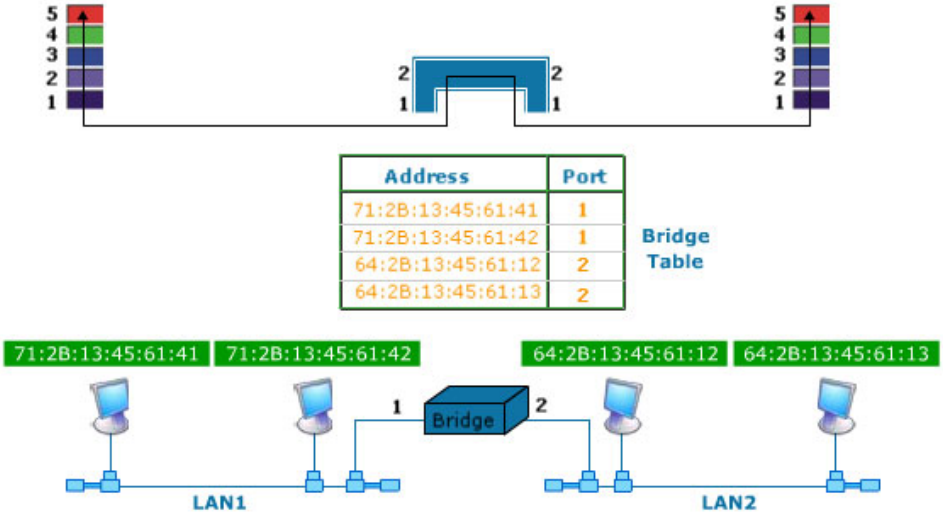
لاحظ أن الاستخدام الهرمي للمجمعات يلغي الطول الأعظمي للكابل المحدد بمائة متر في حالة 10 Base-T؛ لكن يبقى لدينا حد أعلى لعدد المجمعات الممكن ربطها بشكل هرمي.

4.1 الجسر Bridge

يعمل الجسر على المستويين الفيزيائي ووصلة المعطيات معاً. بالنسبة للمستوى الفيزيائي، فالجسر يقوم بإعادة توليد الإشارات المستقبلة؛ أما بالنسبة لمستوى وصلة المعطيات فإن الجسر يستطيع اختبار العناوين الفيزيائية Physical MAC addresses للمصدر وللوجهة الموجودين ضمن الإطار وأخذ القرار المتعلق بتوجيه الإطار أو إهماله؛ أي أن الجسر يستطيع تصفية الأطر Frame Filtering بناءً على العناوين الفيزيائية وفي حال وجود حاجة لتوجيهه فيجب هنا تحديد رقم البوابة التي سيجري التوجيه إليها. فإذاً، يمتلك الجسر جدول تقابل بين العناوين الفيزيائية وأرقام بوابات الجسر المرتبطة بها.

يبين الشكل التالي شبكتين محليتين مربوطتين إلى جسر. فإذا وصل إطار موجه إلى العنوان B13456142712 على البوابة رقم 1 فإن الجسر سيراجع جدول التقابل (أو جدول التوجيه) لتحديد بوابة الانطلاق. فحسب جدول التوجيه المتوفر لدى الجسر فإن الوصول إلى العنوان السابق يجري من خلال البوابة رقم 1 لذلك لا توجد حاجة لإعادة توجيه الإطار الذي يهمل. لكن إذا وصل إطار موجه إلى العنوان B13456141712 على البوابة رقم 2 حيث بوابة الانطلاق هي 1 فيقوم الجسر بإعادة توجيه الإطار على البوابة رقم 1.

ملاحظة: عندما يجري ربط شبكتين محليتين باستخدام جسر فإننا نحصل على شبكة محلية موسعة وليس ترابط شبكات.



ربط شبكتين محليتين بواسطة جسر

هذا يعني أنه في الحالة الأولى تم حصر حركة المرور في الشبكة الأولى بينما في الحالة الثاني فإن حركة المرور امتدت إلى الشبكة الثانية.
طبعاً يمكن للجسر أن يمتلك عدة بوابات في نفس الوقت لكننا نرسم الجسر ببوابتين للتبسيط فقط.
يجب الانتباه هنا إلى أن الجسر لا يغير العناوين الفيزيائية الموجودة ضمن الأطر.

1. الجسور الشفافة Transparent Bridges

تعريف: نقول عن جسر أنه شفاف عندما تكون المحطات غير مدركة لوجوده؛ فلا نحتاج إلى إعادة إعداد المحطات في كل مرة نضيف أو نحذف جسر من الشبكة. تحدد خصائص الجسر الشفاف بما يلي:

- يجب توجيه الأطر من محطة إلى أخرى.
- تُبنى جداول التوجيه وتُحدث آلياً عن طريق التعلم الذاتي من حركة الأطر ضمن الشبكة.
- يجب تجنب حدوث حلقات ضمن الشبكة.

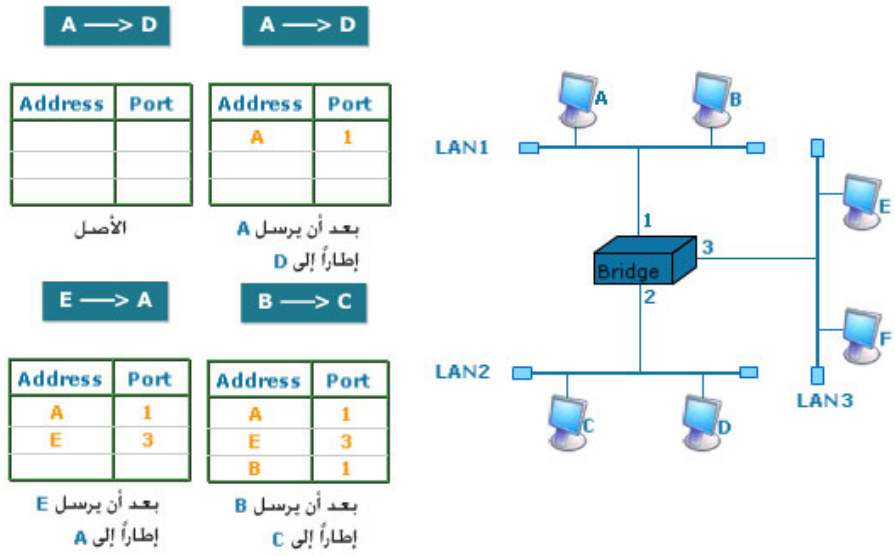
خصائصها حسب المعيار *IEEE 802.1d*

التوجيه

يجب على الجسر الشفاف أن يوجه الأطر توجيهاً صحيحاً كما رأينا سابقاً.

التعلم الذاتي

يمكن تعريف جدول التوجيه الموجود ضمن الجسر بشكل ثابت عن طريق إدخال العناوين الفيزيائية المربوطة إلى كل بوابة لكن تلك الطريقة تبقى غير عملية لأننا في كل مرة نضيف محطة جديدة أو نغير مكان المحطة أو نغير بطاقة الشبكة لمحطة ما يجب علينا أن نعدل جدول التوجيه. لذلك من الأفضل أن يحدث جدول التوجيه آلياً. حتى نستطيع تحقيق ذلك، لا بد للجسر من التعلم من حركة الأطر وذلك عن طريق اختبار عنواني الوجهة والمصدر؛ بناءً على عنوان الوجهة يتم اتخاذ قرار التوجيه أو عدمه بينما يفيد عنوان المصدر لإضافة إدخال جديد إلى جدول التوجيه أو تعديل إدخال موجود مسبقاً.
يساعدنا الشكل التالي على التوسع قليلاً في هذه الفكرة:



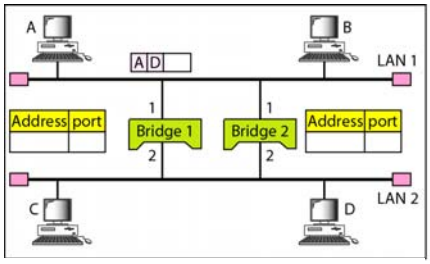
آلية التعلم ضمن جسر شفاف

- عندما ترسل المحطة A إطاراً إلى المحطة D، لا يحوي الجسر أي إدخال Entry لعناوين A أو D لذلك فإنه يقوم بإرسال الإطار على جميع البوابات ما عدا البوابة التي وصل منها الإطار (أي يتم الإرسال على البوابتين 2 و 3). عندما يرى الجسر عنوان المصدر A فإنه يتعلم أنه يمكن الوصول إلى المحطة A عن طريق البوابة رقم 1. هذا يعني أنه في المستقبل يجب على الجسر توجيه الأطر الموجهة إلى A إلى البوابة رقم 1. يضيف الجسر هذا الإدخال إلى جدول التوجيه.
- عندما ترسل المحطة E إطاراً إلى المحطة A، فإن الجسر الذي يحوي إدخالاً لـ A يقوم بتوجيه الإطار مباشرةً إلى البوابة رقم 1 فقط. يضيف الجسر عنوان المصدر (وهو E) إلى جدول التوجيه.
- عندما ترسل المحطة B إطاراً إلى المحطة C، يقوم الجسر بتوجيه الإطار على البوابات 2 و 3 ومن ثم يضيف B إلى جدول التوجيه.
- تستمر عمليات التعلم طالما الجسر يقوم بتوجيه الأطر.

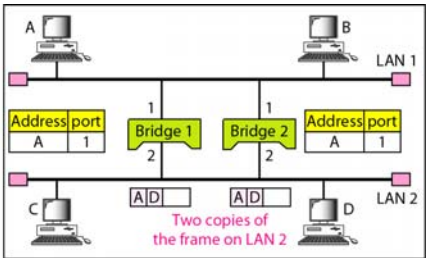
2. <u>مشكلة الحلقة Loop</u>

تعمل الجسور الشفافة بشكل جيد طالما لا توجد جسور مكررة ضمن الشبكة. إلى أن مديرو الأنظمة يفضلون وضع جسور مكررة لوصل شبكتين بغية زيادة الوثوقية. فعندما يتعطل أحد الجسور يستطيع الجسر الآخر تحمل المسؤولية ومتابعة العمل لحين تصليح الجسر المعطل أو تغييره. إن وضع الجسور بشكل مكرر ضمن الشبكة غير مرغوب به لأنه يخلق حلقات.

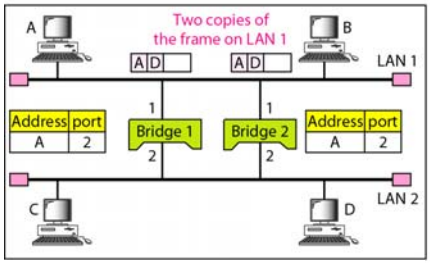
يظهر الشكل التالي مثال عن شبكة تحوي حلقة تم خلقها عن طريق وصل شبكتين بواسطة جسرين.



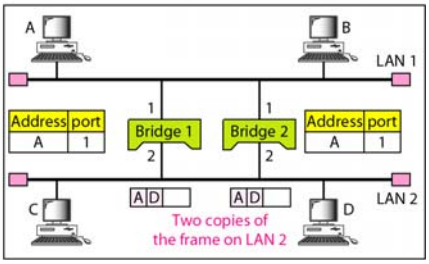
أ – ترسل المحطة A إطاراً إلى المحطة D



ب – كلا الجسرين يوجه الإطار



ج – كلا الجسرين يوجه الإطار



د – كلا الجسرين يوجه الإطار

مشكلة وجود الحلقات في الجسور الشفافة

- ترسل المحطة A الإطار إلى المحطة D. بما أن جداول الجسرين فارغة فيجري توجيه الإطار وإضافة عنوان A إلى جداولهم.
- لدينا الآن نسختين من الإطار على الشبكة LAN2. يستقبل الجسر الثاني الإطار الذي أرسله الجسر الأول وهو لا يعرف إلى الآن أين تقع المحطة D؛ لذلك يقوم الجسر الثاني بإرسال الإطار على البوابة رقم 1. تصل النسخة الأخيرة من الإطار إلى الجسر الأول الذي يعيد إرسالها من جديد لعدم وجود معلومات لديه عن مكان المحطة D. في جميع الحالات، يجري تحديث الجداول الموجودة ضمن الجسور وفق الإطارات القادمة.
- يوجد لدينا الآن نسختين من الإطار على LAN1. يجري إعادة الخطوة (ب) السابقة.
- تستمر العمليات إلى اللانهاية.

لحل هذه المشكلة، يطلب معهد IEEE استخدام خوارزمية شجرة التغطية Spanning Tree Algorithm بغية خلق طبولوجية خالية من الحلقات.

3. شجرة التغطية

تعمل خوارزمية شجرة التغطية بين الجسور فقط بغية تحويل الطبولوجية الفيزيائية التي يمكن أن تحوي حلقات إلى طبولوجية خالية من الحلقات عن طريق تعطيل بعض الجسور عن العمل أو تعطيل بعض بوابات الجسور وفي أي لحظة يتعطل جسر عن العمل فإن خوارزمية شجرة التغطية تعيد آلياً تكوين شجرة تغطية أخرى.

4. استخدام الجسور لربط شبكات محلية مختلفة

يجب أن يكون الجسر، نظرياً، قادراً على ربط شبكتين محليتين مزودتين ببروتوكولات طبقة وصلة المعطيات مختلفة، كربط شبكة إيثرنت مع شبكة WiFi اللاسلكية. توجد، في هذه الحالة بعض المسائل التي يجب مراعاتها:

- صيغة الإطار.

يجب أن يكون الجسر قادراً على تحويل الأطر من صيغة إلى صيغة أخرى وذلك لأن كل شبكة محلية تعرف صيغة خاصة بها.

- الطول الأعظمي للمعطيات Maximum Transfer Unit (MTU).
- عندما يكون طول المعطيات أكبر من الطول المسموح به في الشبكة الواقعة على الطرف الآخر للجسر، فيجب أن يقوم الجسر بتجزئة الإطار إلى عدة أطر على أن تتم إعادة التجميع عند الوجهة. لا يوجد، على أية حال، أي بروتوكول ينتمي إلى طبقة وصلة المعطيات يسمح بتجزئة الأطر أو إعادة تجميعها لذلك يسعى الجسر إلى إهمال الأطر الكبيرة.
- معدل نقل المعطيات.
- تتميز كل شبكة بمعدل نقل معطيات خاص بها. فشبكة إيثرنت تنقل المعطيات بمعدلات 10 Mbps أو 100 Mbps أو 1 Gbps بينما الشبكات اللاسلكية فتنقل المعطيات بمعدلات 1 Mbps أو 11 Mbps أو 54 Mbps، لذلك يجب أن يكون الجسر قادراً على تخزين كمية كبيرة من المعطيات ليستطيع تفادي مشكلة عدم التناظر بالسرعة بين بوابة الدخل وبوابة الخرج.

- ترتيب البتات.

بعض الشبكات ترسل البت الأقل دلالةً أولاً بينما البعض الآخر يرسل البت الأكثر دلالة أولاً؛ لذلك يجب أن يكون الجسر قادراً على التعامل مع هذه المشكلة.

- الأمن.

بعض الشبكات مثل اللاسلكية تحقق مستويات أمنية جيدة ضمن طبقة وصلة المعطيات بينما البعض الآخر لا يعير الجانب الأمني أي انتباه. لذلك يجب أن يكون الجسر قادراً مثلاً على فك تشفير رسالة ما عندما تنتقل من شبكة آمنة إلى شبكة غير آمنة.

- دعم التطبيقات متعددة الوسائط Multimedia Support.

تدعم بعض الشبكات التطبيقات متعددة الوسائط وتؤمن جودة الخدمة Quality Of Service (QoS) المطلوبة لهذه التطبيقات والبعض الآخر لا يؤمنها.

5. مبدلات الطبقة الثانية Layer-2 Switches

يجب هنا التمييز بين نوعين من المبدلات:

النوع الأول: يعمل على المستوى الثاني (أي الطبقتين الفيزيائية ووصلة المعطيات).

النوع الثاني: يعمل على المستوى الثالث (أي الطبقات الفيزيائية ووصلة المعطيات والشبكة).

تعريف: مبدلة الطبقة الثانية هي جسر متعدد البوابات مزود بإمكانيات معالجة متطورة.

يُستخدم الجسر عادةً لربط عدد محدد من الشبكات المحلية، لكن عندما يتوفر لدينا مبدلة بعدد كبير من البوابات فنستطيع ربط المحطة مباشرةً إلى المبدلة أو بشكل آخر تصبح كل شبكة محلية مكونة من المحطة وبوابة المبدلة (لا تصادمات بعد الآن).

تصفي المبدلة مثلها مثل الجسر الأطر حسب العناوين الفيزيائية كما يمكن تزويدها بدارئة لتخزين الأطر لحين معالجتها.

نذكر من الميزات الأخرى التي تتمتع بها المبدلة ما يلي:

- إمكانية توجيه الإطار إلى بوابة الخرج بعد فحص عنوان الوجهة فقط ودون الحاجة إلى إدخال كامل الإطار واختبار الأخطاء عليه.
- تعرف هذه الإمكانية بـ Cut-through switching وهي تحسن أداء المبدلة (زمن تأخير إرسال الإطار هو من مرتبة 6 بايتات) لكن في نفس الوقت تزيد معدل الأخطاء الناتجة إما عن تشوه الإشارة نتيجة الضجيج أو نتيجة التصادم.
- إمكانية توجيه الإطار إلى بوابة الخرج بعد إدخال أول 64 بايت منه.

تعرف هذه الإمكانية باسم Fragment-free cut-through switching وهي تؤخر إرسال الإطار بزمن دخول أول 64 بايت لكن في نفس الوقت تحسن الوثوقية وذلك لأننا بعد مرور أول 64 بايت من الإطار نصبح متأكدين أن الإطار لم يتعرض لتصادم.

- إمكانية وجود بوابة صاعدة أو أكثر تدعى Uplinks.

يكون معدل النقل لهذه البوابة أكبر منه لدى بقية البوابات وتفيد هذه البوابة في ربط مبدلات مع بعضها البعض أو ربط مخدم إلى الشبكة. فعند التكلام عن مبدلة -24) port 10Base-T with 2-port 100Base-Tx uplinks) نقصد مبدلة مزودة بـ /24/ بوابة تعمل بمعدل 10 Mbps وببوابتين تعملان بسرعة 100 Mbps.

2. أجهزة ترابط الشبكات Network Interconnection Devices

1.2 المسيرات Routers

يعمل المسير على المستوى الثالث أي يقوم بتسيير الطرود حسب العنوان المنطقي. يربط المسير عادةً الشبكات المحلية مع الشبكات الواسعة ضمن الإنترنت ويحوي جدول تسيير يساعده على اتخاذ قرارات التسيير اعتماداً على أفضل مسار (أو طريق) بين المصدر والوجهة. تكون جداول التسيير عادةً آلية التأقلم حيث يجري تحديثها باستخدام بروتوكولات التسيير Routing Protocols.

2.2 مبدلات الطبقة الثالثة Layer-3 Switches

تمتاز المبدلة التي تعمل على المستوى الثالث عن مسير بكونها أكثر سرعة وأكثر تطوراً. فالبنية الداخلية للمبدلة Switching Fabric تجعل عمليات البحث ضمن الجداول والتسيير أكثر سرعة.

3.2 العبارة Gateway

تعمل العبارة ضمن جميع المستويات الخمسة بالنسبة للإنترنت والسبعة بالنسبة للنموذج OSI.
تستقبل العبارة رسالة من تطبيق ما وتقرأه وتفسره مما يعني أننا يمكن استخدام العبارة لربط نظامين يستخدمان نموذجين مختلفين. لربط، على سبيل المثال، شبكة نظام يستخدم نموذج OSI المعياري ونظام آخر يستخدم نموذج الإنترنت. تستطيع العبارة الولوج إلى جميع الترويسات التي تولدها كل الطبقات وتوليد ترويسات مختلفة لذلك يطلق عليها في بعض الأحيان اسم محول بروتوكولات Protocol Converter.
يمكن أخيراً للعبارة أن تؤمن الحماية والأمن عن طريق تصفية الرسائل على مستوى التطبيقات Application-level filtering.

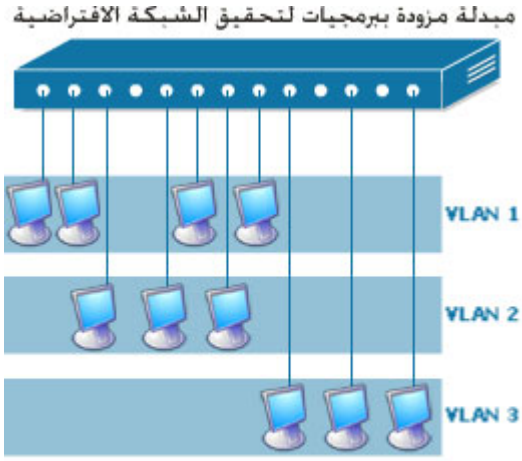
3. الشبكات المحلية الافتراضية VLANS

تعتبر محطة ما جزءاً من شبكة محلية إذا كانت فيزيائياً مربوطة إلى تلك الشبكة، لذلك فعلاقة الانتماء هي علاقة جغرافية.
تعريف: الشبكة المحلية الافتراضية Virtual LAN (VLAN) هي شبكة محلية معرفة عن طريق البرمجيات وليس الطبولوجية الفيزيائية.
لنأخذ المثال المبين في الشكل التالي:

- ربط شبكات محلية عن طريق مبدلة -

لاحظ أننا مددنا الشبكة المحلية بطريقة تسمح بتعريف ثلاث مجموعات عمل بحيث يعمل أعضاء كل مجموعة سويةً أو يكون لديهم مشاريع مشتركة. لكن ماذا يحصل لو أراد مدير النظام نقل مهندسين من المجموعة الأولى إلى المجموعة الثالثة لتسريع العمل مثلاً؟ يجب على التقنيين إعادة تمديد الكابلات من جديد ومن ثم إعادة التمديد في كل مرة يطرأ تعديل على عضوية كل مجموعة.

يظهر الشكل التالي نفس الشبكات المحلية مقسمة إلى شبكات افتراضية.



- تزويد المبدلة ببرمجيات
لتحقيق الشبكة الافتراضية -

تكمن الفكرة الأساسية خلف الشبكات الافتراضية بتقسيم الشبكة منطقياً وليس فيزيائيا. يدعى كل قسم منطقي بشبكة افتراضية وكل شبكة افتراضية تمثل مجموعة عمل أو قسم مختلف ضمن المؤسسة. لا نحتاج في هذه الحالة إلى تغيير التمديدات الفيزيائية للتأقلم مع انتقال الأشخاص من مجموعة عمل إلى أخرى وذلك لأننا نحدد عضوية الأشخاص إلى المجموعات عن طريق البرامج. لذلك نستطيع نقل كل محطة منطقياً من شبكة افتراضية إلى شبكة افتراضية أخرى.

عندما يرسل عضو من شبكة افتراضية تعميم فإن أعضاء هذه الشبكة الافتراضية فقط هم الذين يستقبلون الرسالة. أي أن الشبكات الافتراضية تسمح بتحسين أداء الشبكة عن طريق حصر رسائل التعميم على مستوى الشبكة الافتراضية وليس الشبكة ككل.

الوحدة السابعة
العنونة المنطقية لبروتوكول IPv4

سنناقش في هذه الوحدة مفاهيم العنونة والشبكات الجزئية المستخدمة ضمن الإنترنت. نستخدم بكثرة مصطلح عنوان IP للدلالة على العنوان المنطقي لجهاز أو لمضيف Host كما جرت العادة على تسميته ضمن شبكة الإنترنت.

1.عناوين IPv4

يتكون عنوان IP من 32 bits وهو يعرف تعريفاً وحيداً على مستوى العالم بأسره.
نقصد بكون عنوان IP وحيداً أنه يعرف اتصالاً وحيداً بالإنترنت. لا يمكن لجهازين واقعين على الإنترنت امتلاك نفس عنوان IP في نفس الوقت. طبعاً يمكن لجهاز امتلاك عنوان IP لفترة محددة من الوقت، كفترة اتصاله بالإنترنت، ومن ثم تحرير العنوان لجهاز آخر.
إذا احتوى جهاز ما على عدة وصلات مع الإنترنت فيجب تخصيص عنوان IP لكل وصلة.

1.1 فضاء العنونة

تعريف: فضاء العنونة هو المجموع الكلي للعناوين التي يستخدمها بروتوكول IPv4.
إذا استخدم بروتوكولاً ما N bits لتعريف العنوان فإن فضاء العنونة يكون 2^N.
يستخدم الإصدار الرابع من بروتوكول IP 32 بت للعنونة مما يعني أننا نستطيع عنونة 2^{32} جهازاً أو 4,294,967,296. لكن لم يجر استخدام فضاء العنونة هذا بشكل جيد بسبب بعض القيود الموضوعة عليه الأمر الذي أدى إلى هدر جزء كبير من العناوين.

1.1.1 التدوين Notation

يمكنك تدوين عنوان IP باستخدام طريقتين:

- التدوين الثنائي Binary

يجري هنا كتابة العنوان على شكل 32 بت (أو أربعة بايتات) مثل العنوان التالي:

01110101 10010101 00011101 00000010

- التدوين العشري المنقط Dotted-decimal Notation

يفيد التدوين العشري في جعل العنوان أقل حجماً وأسهل للقراءة والحفظ. نكتب هنا كل بايت من البايتات الأربعة المكونة للعنوان بالشكل العشري مع إضافة نقطة بين البايتات مثل العنوان التالي:

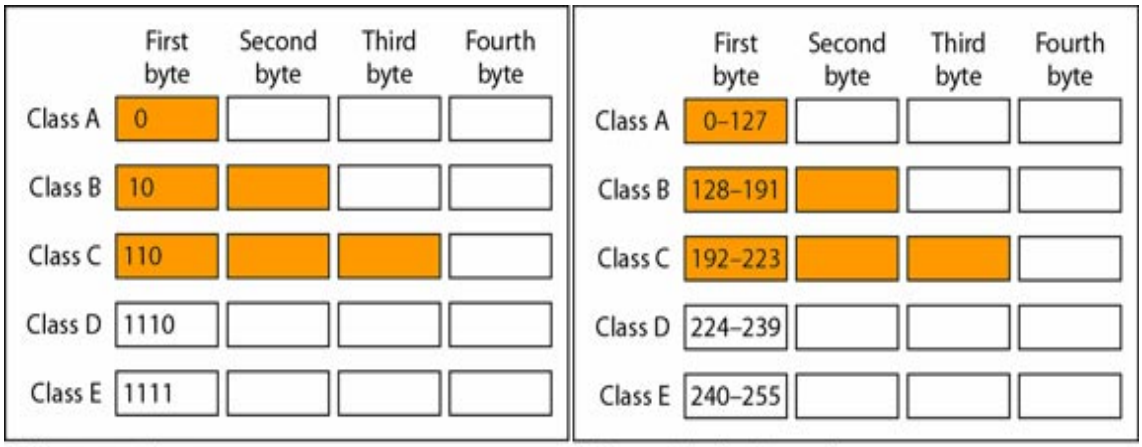
117.149.29.2

2. العنونة الصفّية Classful Addressing

سنناقش بإيجاز عناوين IP الصفّية (التي توزع إلى عدة صفوف) وذلك لأنها تخرج من الخدمة تدريجياً. يكون عنوان IP هنا موزعاً على خمسة صفوف: A, B, C, D, and E. يشغل كل صف جزءاً من فضاء العنونة المتاح.

يمكننا معرفة الصف الذي ينتمي إليه عنوان ما من خلال البتات الأولى للعنوان (من اليسار) إذا كان مدوناً بالصيغة الثنائية أو من خلال البايت الأول من اليسار إذا كان مدوناً بالصيغة العشرية.

يبين الشكل التالي صفوف العنونة المستخدمة وبتات أو بايتات الدلالة على نوع الصف.



a. Binary notation

b. Dotted-decimal notation

ـ إيجاد صفوف العناوين وفق التدوين الثنائي أو العشري ـ

تكمن المشكلة مع هذا النوع من العناوين في كونها مقسمة إلى أعداد ثابتة من الكتل وكل كتلة مؤلفة من حجم ثابت أيضاً.

الصف	عدد الكتل	حجم الكتلة	الاستخدام
A	128	128	وحيد الوجهة Unicast
B	65,536	16,384	وحيد الوجهة
C	256	2,097,152	وحيد الوجهة
D	268,435,456	1	متعدد الوجهات Multicast
E	268,435,456	1	محجوز

جدول 1 – عدد الكتل وحجمها ضمن صفوف عناوين IPv4

عندما تطلب أي مؤسسة كتلة عناوين فكانت تخصص بكتلة تنتمي إلى أحد الصفوف A, B, or C وذلك حسب حجم المؤسسة. لاحظ أن الصف A يحوي عناوين تزيد عن حاجة أكبر الشركات مما يعني أنه يجري هدر معظم هذه العناوين. كذلك الأمر بالنسبة للصف B الذي يزيد عن حاجة الكثير من المؤسسات؛ فهنا أيضاً يوجد هدر. بالنسبة للصف C فهو أصغر من حاجة معظم المؤسسات. الصف D مخصص للإرسال متعدد الوجهات والصف E مخصص للتجارب.

1.2 معرف الشبكة ومعرف المضيف Netid and Hostid

يجري تقسيم عنوان IP في الصفوف A, B, and C فقط إلى معرف شبكة ومعرف مضيف ذات أطوال متغيرة ومتعلقة بصف العناوين. يبين الجدول التالي بعض البايتات المخصصة لمعرف الشبكة باللون الأزرق ولمعرف المضيف.

الصف	التمثيل الثنائي لقناع الشبكة	التمثيل العشري لقناع الشبكة	CIDR
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

الجدول 2 – قناع الشبكة الافتراضي للعناوين المصفة

لاحظ أننا نستخدم ضمن الصف A بايتاً واحداً للدلالة على معرف الشبكة وثلاثة بايتات للدلالة على معرف المضيف (المحطة) داخل الشبكة. بينما نستخدم، ضمن الصف B، بايتين للدلالة على معرف الشبكة وبايتين للدلالة على معرف المضيف.

ونخصص أخيراً ثلاثة بايتات لمعرف الشبكة وبايت واحد لمعرف المضيف ضمن الصف C.

2.2 مفهوم قناع الشبكة Net Mask

مع العلم أن الأقسام المخصصة لمعرف الشبكة وتلك المخصصة لمعرف المضيف محددة تحديداً ثابتاً إلا أننا نستطيع أن نستخدم مفهوم قناع الشبكة الذي يتكون من 32 بت. يبين الجدول 2 قناع الشبكة الافتراضي المستخدم لصفوف العناوين A, B, and C فقط.

يفيد القناع في إيجاد الجزء المخصص لمعرف الشبكة أو لمعرف المضيف من عنوان IP ما.

فمثلاً بما أن قناع الشبكة للصف A مكون من ثمانية واحدات متتالية فهذا يعني أن أول بايت من

العنوان هو عنوان الشبكة (أي أن الواحدات المتتالية تحدد معرف الشبكة والأصفار المتتالية تحدد معرف المضيف).

يظهر العمود الأخير من الجدول القناع مكتوباً بالصيغة /n حيث يأخذ المتحول n القيم: 8, 16, or 24. يدعى هذا التدوين Slash Notation أو Classless Interdomain Routing (CIDR). يستخدم هذا التدوين في عناوين IP اللاصفية الذي سنناقشها لاحقاً.

3.2 الشبكات الجزئية Subnetting

خلال فترة استخدام العناوين الصفّية، كان من الممكن إدخال مستوٍ ثالث من الهرمية عن طريق معرف الشبكة الجزئية Subnetid. يجري تحقيق ذلك عن طريق زيادة عدد الواحدات المعرفة لرقم الشبكة بحيث يدل العدد الإضافي من الواحدات على معرف الشبكة الجزئية. بذلك، تستطيع أي مؤسسة كبيرة أن تقسم العناوين إلى عدة مجموعات وتعطي لكل مجموعة معرف شبكة جزئية خاص بها.

3. العنونة اللاصفّية Classless Addressing

تفيد العنونة اللاصفية بتجاوز القيود التي وضعتها العنونة الصفية وبالمقدرة على توزيع عدد أكبر من الكتل ومن العناوين. لا توجد، في هذه الحالة، أية صفوف عنونة بينما نحافظ على آلية منح العناوين عن طريق الكتل.

1.3 كتل العناوين Address Blocks

عندما تريد مؤسسة ما الاتصال بالإنترنت، فإنه يتم منحها كتلة (أو مجال) عناوين. يتعلق حجم الكتلة (أي عدد العناوين الممنوحة) بحجم المؤسسة وطبيعة عملها. فيمكننا تخصيص الحواسيب المنزلية بكتلة مؤلفة من عنوانين فقط بينما تخصص مؤسسة كبيرة مثل مزود خدمات الإنترنت Internet Service Provider (ISP) بعدة آلاف من العناوين حسب حاجته.

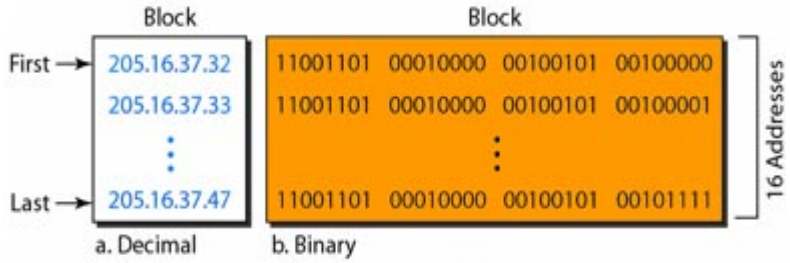
القيود الموضوعة:

لتسهيل عملية توزيع العناوين، تضع سلطات الإنترنت القيود التالية على كتل العناوين اللاصفية:

1. يجب أن تكون العناوين ضمن الكتلة متسلسلة (أي كل عنوان يلي العنوان السابق).
2. يجب أن يكون عدد عناوين كتلة ما من أس 2 (1, 2, 4, 8, …).
3. يجب أن يقبل العنوان الأول من الكتلة القسمة على عدد العناوين المخصصة.

مثال 1

يبين الشكل التالي كتلة عناوين بالتدوين الثنائي والعشري ممنوحة إلى شركة تحتاج 16 عنوان.



كتلة مكونة من 16 عنوان ممنوحة لشركة صغيرة

لاحظ أننا احترمنا القيود الثلاثة السابقة؛ العناوين متسلسلة وعـدد العنـاوين هـو 16 (أي 2^4) والعنوان الأول يقبل القسمة على 16.

2.3 قناع الشبكة Mask

تعتمد أفضل طريقة لتحديد كتلة عناوين على اختيار عنوان ما من الكتلة والقناع. كما قلنا سابقاً، يتألف القناع من 32 بت لكن الاختلاف الوحيد هنا أن عدد الواحدات ضمن القناع يمكن أن يتدرج من الصفر إلى 32. لذلك نستعيض عن القناع بالتدوين /n حيث تدل n على عدد الواحدات ضمن القناع.

نتيجة: يمكن تعريف كتلة عناوين ضمن عناوين IPv4 اللاصفية وفق

x.y.z.t/n

حيث تعرف القيمة x.y.z.t أحد العناوين ويحدد /n القناع

لاحظ أن العنوان مع /n يحددان جميع المعلومات المطلوبة من الكتلة (أي عدد العناوين وأول عنوان وآخر عنوان ضمن الكتلة).

- **أول عنوان داخل الكتلة**

يمكن حساب أول عنوان داخل الكتلة عن طريق تصفير بتات أقصى اليمين الـ 32-n وفق التدوين الثنائي.

مثال 2

جرى منح شركة صغيرة كتلة عناوين. نعرف أن أحد هذه العناوين هو 205.16.37.39/28.

ما هو أول عنوان ضمن الكتلة؟

إيضاح: يظهر الحل بعد النقر على زر الجواب.

الحل

التمثيل الثنائي للعنوان هو 11001101 00010000 00100101 00100111.

نصفر البتات اليمينية الـ 28 – 32 (أي الأربعة الأولى) فيصبح العنوان هو

11001101 00010000 00100101 00100000

التمثيل العشري للعنوان الثنائي السابق هو: 205.16.37.32 وهو أول عنوان ضمن الكتلة المطابق للشكل العلوي.

- **آخر عنوان داخل الكتلة**

نحصل على آخر عنوان ضمن الكتلة عن طريق وضع قيمة البتات الـ 32-n اليمينية تساوي الواحد.

تمرين: احسب آخر عنوان ضمن المثال السابق.

- **عدد العناوين ضمن الكتلة**

وهو الفرق بين آخر عنوان وأول عنوان أو يمكن حسابه مباشرةً من الصيغة 2^{32-n}.

عنوان الشبكة

يجري عادةً تخصيص أول عنوان ضمن الكتلة كعنوان خاص يدل على الشبكة نفسها ولا يمنح لأي حاسب. يدل هذا العنوان على المؤسسة نفسها ويفيد في التعريف عنها ضمن الإنترنت.

3.3 الهرمية

- **الهرمية من المستوى الثاني (بدون شبكات جزئية):**

يستطيع عنوان IP أن يعرف مستويين هرميين عندما لا نستخدم التجزئة. فتعرف البتات n من أقصى اليسار x.y.z.t/n الشبكة وتعرف بقية البتات اليمينيـة (أي 32-n) المضيف. يدعى معرف الشبكة بالسابقة Prefix ومعرف المضيف باللاحقة Suffix.

- **الهرمية من المستوى الثالث (مع التجزئة):**

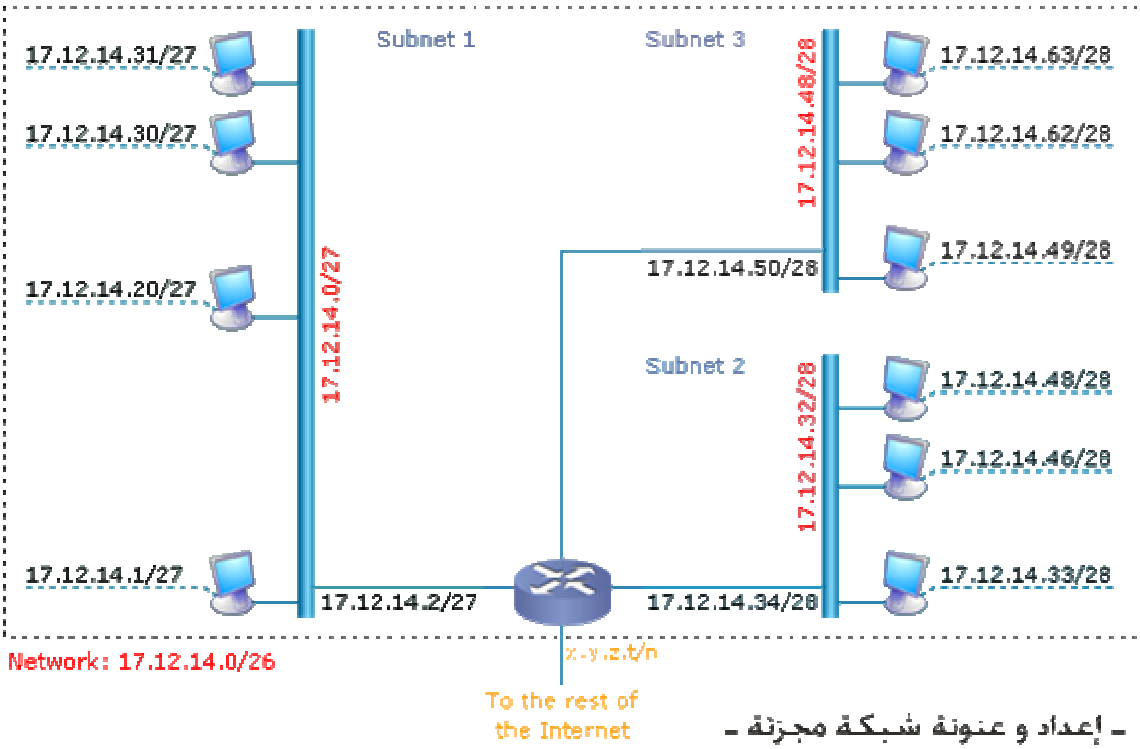
يمكن أن تلجأ أي شركة منحت كتلة كبيرة من العناوين إلى تجزئة هذه الكتلة إلى مجموعة من الشبكات الجزئية Subnets وتوزيع العناوين بين هذه الشبكات الجزئية. يجري هذا التقسيم بطريقة شفافة بالنسبة للعالم الخارجي الذي يظل يرى هذه الشركة كأنها مكونة من شبكة واحدة لكن داخلياً يوجد عدة شبكات جزئية. ترسل جميع الرسائل إلى عنوان المسير الذي يحقق عملية ربط الشركة ببقية الإنترنت؛ يقوم المسير بتسيير الرسائل إلى الشبكة الجزئية المناسبة. يتوجب على الشركة، لتحقيق ذلك، خلق كتل جزئية وتوزيعها على

الشبكات الجزئية. تمتلك الشركة قناع خاص بها كذلك سوف تمتلك كل شبكة جزئية قناع آخر خاص بها.

لنفترض أنه تم منح شركة ما كتلة العناوين 17.12.40.0/26، أي ما مقداره 64 عنوان. لنفترض أيضاً أن الشركة مكونة من ثلاثة مكاتب وأنها تحتاج إلى تقسيم الكتلة السابقة إلى 3 كتل جزئية تحوي على التوالي: 32 و16 و16 عنواناً. يمكننا إيجاد القناع الجديد كما يلي:

- بافتراض أن قناع الشبكة الجزئية الأولى هو n1، ينـتج عنـه أن $2^{32-n1} = 32$. أي أن n1=27.
- بنفس الطريقة نستنتج أن n2=28 و n3=28.

بالنتيجة، يصبح لدينا القناع 26 للشركة وثلاثة أقنعة لكل شبكة جزئية كما هو موضح بالشكل التالي.



إعداد و عنونة شبكة مجزئة ـ

إعداد وعنونة شبكة مجزئة

لنرى إذا كان بإمكاننا إيجاد عناوين الشبكة الجزئية من خلال أحد عناوين الأجهزة الموجودة ضمنها.

- في الشبكة الجزئية Subnet1، إن العنوان 17.12.14.29/27 يمكن أن يعطينا عنوان الشبكة الجزئية إذا استخدمنا القناع /27 وذلك لأن:

Host : 00010001 00001100 00001110 00011101

Mask: /27
Subnet: 00010001 00001100 00001110 00000000 → (17.12.14.0)

- في الشبكة الجزئية Subnet2، إن العنوان 17.12.14.45/28 يمكن أن يعطينا عنوان الشبكة الجزئية إذا استخدمنا القناع /28 وذلك لأن:

Host : 00010001 00001100 00001110 00101101
Mask: /28
Subnet: 00010001 00001100 00001110 00100000 → (17.12.14.32)

- في الشبكة الجزئية Subnet3، إن العنوان 17.12.14.50/28 يمكن أن يعطينا عنوان الشبكة الجزئية إذا استخدمنا القناع /28 وذلك لأن:

Host : 00010001 00001100 00001110 00110010
Mask: /28
Subnet: 00010001 00001100 00001110 00110000 → (17.12.14.48)

تمرين :

تحقق أن تطبيق القناع /26 على أي عنوان من العناوين يعطينا عنوان الشبكة 17.12.14.0/26.

يمكننا الآن القول أنه باستخدام الشبكات الجزئية أصبح لدينا ثلاثة مستويات من الهرمية؛ سابقتين للشبكة وللشبكة الجزئية ولاحقة للمضيف.

1.3.3 هرمية متعددة المستويات

لا تضع بنية العناوين اللاصفية أية قيود على عدد المستويات الهرمية. فيمكن، على سبيل المثال، لمزود خدمات إنترنت وطني أن يقسم كتلة العناوين الممنوحة له إلى مجموعة من الكتل الجزئية ويمنح كل كتلة إلى مزود إنترنت إقليمي الذي يقوم بدوره بتجزئة العناوين الممنوحة له إلى مجموعات جزئية ومنحها لمزودي إنترنت محليين. يقوم مزود الإنترنت المحلي بتقسيم كتلة العناوين الممنوحة له ويوزعها على المؤسسات. في النهاية، تستطيع أي مؤسسة تقسيم كتلة العنونة على شبكات جزئية.

2.3.3 تخصيص العناوين

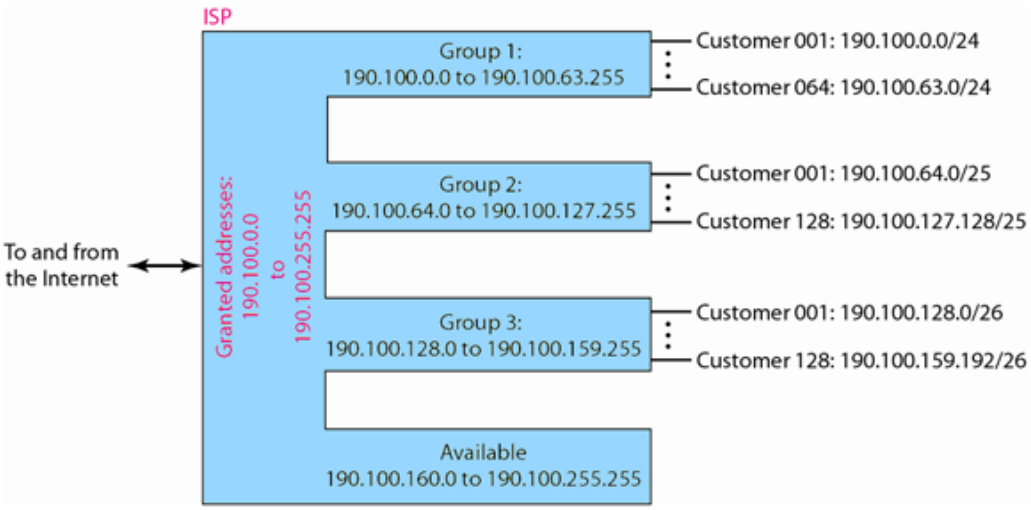
لكن من هي الجهة المسؤولة عن تخصيص العناوين؟ لقد جرى منح هذه المسؤولية إلى سلطة عامة تـدعى Internet Corporation for Assigned Names and Addresses (ICANN). إلا أن هذه السلطة لا تخصص العناوين إلى المؤسسات مباشرةً بل تخصص كتل ضخمة إلى مزودي خدمة الإنترنت. يقوم هنا كل مزود خدمة إنترنت بتقسيم الكتلة الممنوحة له وتوزيعها على المستثمرين.

مثال 3

منح مزود خدمة إنترنت كتلة العناوين 190.100.0.0/16 أي 65,536 عنوان. يحتاج مزود الخدمة هذا إلى تخصيص كتلة العناوين هذه إلى المجموعات التالية:

- تضم المجموعة الأولى 64 زبون؛ كل زبون يحتاج 256 عنواناً.
- تضم المجموعة الثانية 128 زبون؛ كل زبون يحتاج 128 عنواناً.
- تضم المجموعة الثالثة 128 زبون؛ كل زبون يحتاج 64 عنواناً.

صمم الكتل الجزئية وبين أنه ما زال يوجد عناوين متاحة.



مثال عن تخصيص عناوين مزود خدمة إنترنت

1. المجموعة الأولى

بما أن كل زبون يحتاج 256 عنواناً فإننا بحاجة إلى 8 بتات لعنونة الأجهزة. سيكون طول السابقة 24 = 8-32. أما العناوين فهي:

الزبون الأول 190.100.0.0/24 إلى 190.100.0.255/24

الزبون الثاني 190.100.1.0/24 إلى 190.100.1.255/24

.....

.....

.....

الزبون رقم 64 190.100.63.0/24 إلى 190.100.63.255/24

المجموع الكلي للعناوين المخصصة : 64 x 256 = 16,384

2. المجموعة الثانية

بما أن كل زبون يحتاج 128 عنواناً فإننا بحاجة إلى 7 بتات لعنونة الأجهزة. سيكون طول السابقة 25 = 7-32. أما العناوين فهي:

الزبون الأول 190.100.64.0/25 إلى 190.100.64.127/25

الزبون الثاني 190.100.64.128/25 إلى 190.100.64.255/25

.....

.....

.....

الزبون رقم 190.100.127.128/25128 إلى 190.100.127.255/25

المجموع الكلي للعناوين المخصصة : 16,384 = 128 x 128

3. المجموعة الثالثة

بما أن كل زبون يحتاج 64 عنواناً فإننا بحاجة إلى 6 بتات لعنونة الأجهزة. سيكون طول السابقة 26 = 6-32. أما العناوين فهي:

الزبون الأول 190.100.128.0/26 إلى 190.100.128.63/26

الزبون الثاني 190.100.128.64/26 إلى 190.100.128.127/26

.....

.....

.....

الزبون رقم 190.100.159.192/26128 إلى 190.100.159.255/26

المجموع الكلي للعناوين المخصصة : 8,1924 = 64 x 128

عدد العناوين الممنوحة إلى مزود الخدمة: 65,535

عدد العناوين التي خصصها مزود الخدمة: 40,960

عدد العناوين المتاحة : 24,576

الوحدة الثامنة

التوجيه والتسيير Forwarding and Routing

لا نحتاج في الشبكات المكونة من شبكة فيزيائية واحدة إلى التسيير أو التوجيه وذلك لأن طبقة وصلة المعطيات Data-Link layer تؤمن تسليم الطرد من المرسل إلى المستقبل ضمن نفس الشبكة الفيزيائية. لكن عندما نتكلم عن مجموعة مترابطة من الشبكات فإنه يتوجب على المرسل أن يختبر عنوان الوجهة، فإذا كان محلياً فإنه يعتمد على طبقة وصلة المعطيات لتسليمه أما إذا كان عنوان الوجهة يقع على شبكة بعيدة فيجب على المرسل تسليمه إلى المسير الافتراضي Default Router الذي يقوم بدوره بتوصيل الطرد إلى وجهته النهائية اعتماداً على جدول تسيير موجود داخله.

1. التسليم Delivery

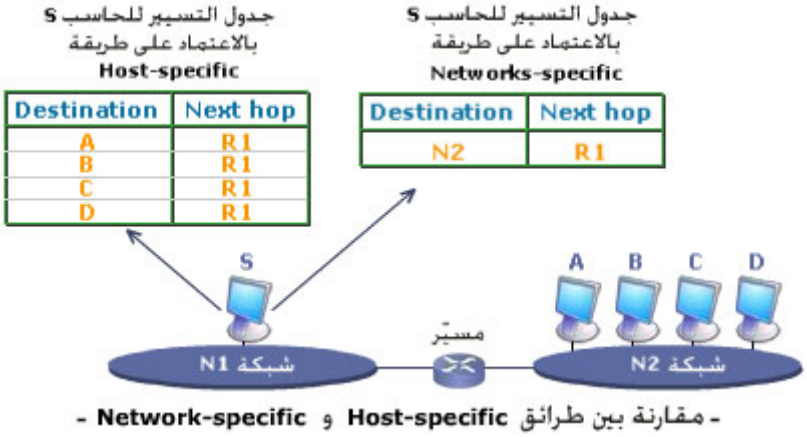
يمكننا التمييز بين نوعين من التسليم:

- التسليم المباشر Direct Delivery: حيث تكون الوجهة النهائية واقعة ضمن نفس الشبكة الفيزيائية للموصل Deliverer أو بشكل آخر إما عندما يقع المرسل والمستقبل على نفس الشبكة الفيزيائية أو بين المسير الأخير والمستقبل ضمن نفس الشبكة النهائية.
  يستطيع المرسل أن يحدد كون التسليم مباشراً أو لا عن طريق مقارنة عنوان الشبكة التي يقع عليها مع عنوان الشبكة للوجهة وفي حال التطابق يكون التسليم مباشراً.
- التسليم غير المباشر Indirect Delivery: ففيها لا يقع المرسل أو المستقبل ضمن نفس الشبكة الفيزيائية الأمر الذي يستدعي مرور الطرد من مسير إلى آخر للوصول إلى الوجهة.
  لاحظ هنا أن التوصيل من المسير النهائي إلى المستقبل يكون مباشراً.

2. التوجيه Forwarding

نقصد بالتوجيه وضع الطرد في المسار المناسب للوصول إلى وجهته النهائية. حتى يستطيع حاسب أو مسير توجيه طرد ما يجب أن يمتلك جدول تسيير Routing Table . فعندما يريد حاسب إرسال طرداً إلى وجهة ما أو عندما يريد مسير Router توجيه طرداً فما عليهما سوى النظر ضمن جدول التسيير لإيجاد الطريق المناسب.
يوجد حالياً عدة طرق لتحقيق التوجيه؛ منها ما يعتمد على وضع المسار كاملاً من المصدر إلى الوجهة ضمن جدول التسيير ومنها ما يضع عنوان القفزة التالية Next hop فقط. توجد أيضاً طرق تسمح بتحديد طريقاً لكل عنوان وجهة وهو ما يعرف بـ Host-specific أو طريقاً لكل

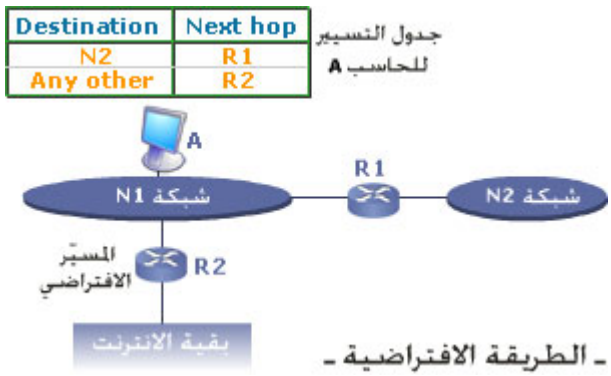
عنوان شبكة الذي تقع عليه الوجهة وهو ما يعرف بـ Network-specific (أي أن المسار يكون واحداً لجميع الحواسب الواقعة ضمن نفس الشبكة الفيزيائية).



لاحظ هنا أن عدد المدخلات ضمن جدول التسيير وحجم كل مدخل يختلف من طريقة إلى أخرى.

1.2 الطريقة الافتراضية Default Method

يبين الشكل التالي حاسب موصول إلى شبكة تحوي مسيرين. يقوم المسير R1 بنقل الطرود إلى حواسب الشبكة N1 بينما يقوم المسير R2 بنقل الطرود إلى بقية الإنترنت. فبدلاً عن إضافة مداخل لكل شبكة من شبكات الإنترنت ضمن جدول التسيير الخاص بالحاسب فإنه يقوم بتوجيه الطرود إلى المسير الافتراضي الذي لا يشغل سوى مدخل واحد ضمن جدول التسيير (عادةً يستخدم عنوان الشبكة 0.0.0.0 للدلالة على بقية الشبكات).



2.2 جداول التسيير Routing Tables

يمتلك كل حاسب مضيف Host أو مسير Router جدول تسيير يحتوي على مدخل لكل وجهة ممكنة أو مجموعة وجهات لمساعدته على تسيير الطرود. يمكن أن يكون جدول التسيير ساكناً Static أو ديناميكياً Dynamic.

1.2.2 الجداول الساكنة

يحوي جدول التسيير الساكن على مسارات يقوم مدير النظام بإدخالها يدوياً، الأمر الذي يضطر مدير النظام إلى تحديث الجدول في كل مرة يطرأ أي تغيير في بنية الشبكات المترابطة. يمكن استخدام هذا النوع من الجداول في ترابط الشبكات الصغيرة التي لا يطرأ عليها تغيرات بكثرة أو ضمن ترابط الشبكات التجريبية بغية اكتشاف وعزل الأعطال.

2.2.2 الجداول الديناميكية

أما بالنسبة لجدول التسيير الديناميكي فيتم تحديثه آلياً باستخدام بروتوكولات التسيير الديناميكية مثل بروتوكول تسيير المعلومات Routing Information protocol (RIP) والبروتوكول المفتوح لإيجاد أقصر الطرق أولاً Open Shortest Path First (OSPF). فعندما يطرأ أي تغيير ضمن الشبكة المترابطة networks Interconnection مثل توقف مسير عن العمل أو انقطاع وصلة ما فإن بروتوكول التسيير يقوم بتحديث جداول التسيير ضمن المسيرات (وفي بعض الحالات عند الحواسب) دون تدخل مدير الشبكة.

3.2 صيغة جدول التسيير

يحوي جدول التسيير أربعة حقول على الأقل وهي: قناع الشبكة Mask وعنوان الشبكة وعنوان القفزة التالية Next Hop وبوابة الخرج Interface. يفيد قناع الشبكة وعنوان الشبكة الوجهة بتعريف عنوان الشبكة التي سيجري تسليم الطرد إليها في آخر قفزة. أما عنوان القفزة التالية فيحدد عنوان المسير التالي ضمن أفضل مسار إلى الوجهة النهائية. أخيراً، تفيد بوابة الخرج في تحديد البوابة التي من خلالها سيجري توجيه الطرد إلى عنوان القفزة التالية.

3. بروتوكولات التسيير Routing Protocols

تعريف: بروتوكول التسيير هو مجموعة من القواعد والإجراءات التي تسمح للمسيرات الموجودة ضمن شبكات مترابطة internet بإعلام بعضها البعض عن التغييرات التي تطرأ على الشبكة.

توضيح: يجري استخدام المصطلح internet التي تبدأ بحرف صغير للدلالة على أي نوع من الشبكات المترابطة حتى وإن لم تكن موصولة إلى شبكة الإنترنت العالمية، بينما يجري استخدام مصطلح Internet التي تبدأ بحرف كبير للدلالة على شبكة الإنترنت المعروفة.

1.3 الاستمثال Optimization

تتجلى الوظيفة الأساسية للمسير في استقبال طرد من بوابة ما وإعادة إرساله إلى بوابة أخرى علماً أنه يمكن أن يحوي المسير عدة بوابات (أي أن يكون موصولاً إلى عدة شبكات في الآن نفسه).

إلى أي بوابة يجب على المسير إرسال الطرد؟

يعتمد الجواب على الاستمثال: اختيار المسار الأفضل (المثالي) إلى الوجهة.

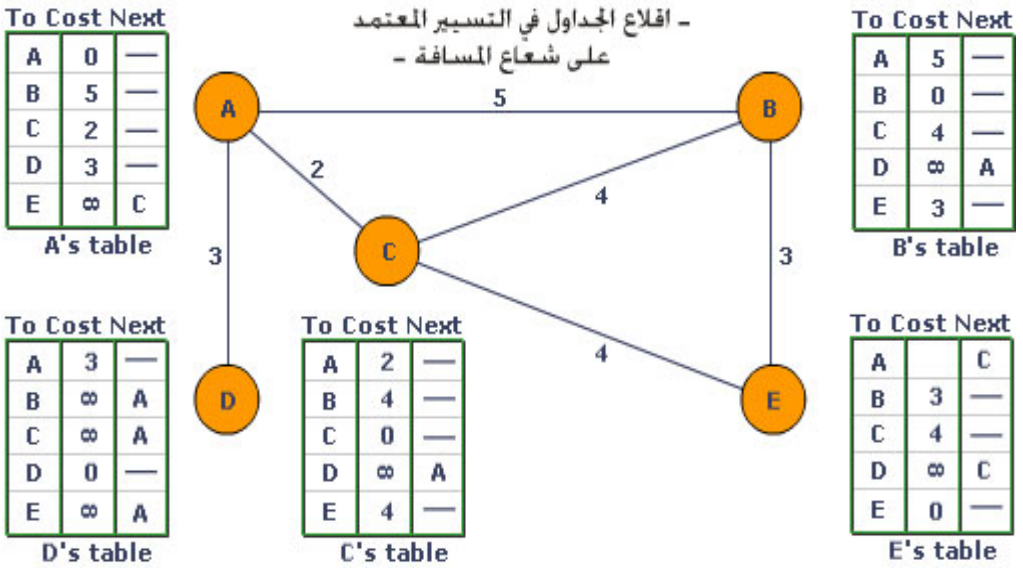
ولتحديد المسار المثالي يجب علينا تحديد سعر cost لكل شبكة يجري المرور عبرها (يطلق أيضاً على السعر اسم المقياس metric). إلى أن المقياس المحدد لكل شبكة يعتمد على البروتوكول المستخدم، فمثلاً يعامل بروتوكول RIP جميع الشبكات بشكل متساوٍ أو أن سعر المرور عبر شبكة ما يأخذ القيمة 1. فإذا مر الطرد عبر عشر شبكات للوصول إلى وجهته النهائية فإن السعر الكلي للانتقال سيكون مساوٍ للقيمة 10 قفزات.

هناك بعض البروتوكولات مثل OSPF التي تسمح بتحديد سعر لكل شبكة حسب مقاييس مختلفة مثل معدل النقل أو جودة الخدمة المتاحة أو زمن التأخير.

2.3 التسيير المعتمد على شعاع المسافة Distance Vector Routing

باستخدام التسيير المعتمد على شعاع المسافة يكون أفضل مسار بين عقدتين هو المسار ذو المسافة الأصغر. تقوم كل عقدة ضمن هذا النوع من البروتوكولات باستخدام شعاع (جدول) يحوي على أقصر المسافات إلى بقية العقد. يفيد الجدول أيضاً في إرشاد العقدة إلى عنوان القفزة التالية (أو الموقف التالي) للوصول إلى الوجهة.

يمكننا تشبيه هذا البروتوكول بالخريطة التي يستخدمها السائحون للانتقال من مدينة إلى مدينة أخرى؛ فالمدن تصبح هي المسيرات والطرق بين المدن تصبح الشبكات والخريطة تصبح الجدول الذي يحوي أقصر المسافات بين أي مدينتين وما اسم المدينة التالية للانتقال بين مدينتين. يبين الشكل مثالاً عن جداول التسيير الموجودة ضمن المسيرات.



جداول التسيير لشعاع المسافة

لاحظ أن جدول المسير A يظهر كيف يمكننا الوصول إلى أي عقدة أخرى من خلال A. فحتى نصل مثلاً من A إلى العقدة E فإن سعر الانتقال الأمثلي هو 6 وعنوان القفزة التالية سيكون هو C.

1.2.3 الإقلاع Initialization

لاحظ أن الجداول الموضحة في الشكل السابق هي مستقرة، فكل عقدة تعرف كيف توجه الطرود إلى أي عقدة أخرى. لكن كيف يمكن للعقدة أن تبني هذا الجدول وتحافظ عليه وهي لا تستطيع أن تعرف إلا المسافة التي تفصلها عن الجيران المباشـرين immediate neighbors، أي الموصولين إليها وصلاً مباشراً. لذلك سنفترض أن كل عقدة تستطيع أن تراسل جيرانها المباشرين وتعرف منهم المسافة التي تفصلها عنهم.

يوضح الشكل التالي الجداول البدائية الموجودة ضمن كل عقدة بعد التراسل مع الجيران المباشرين.

لاحظ أن المسافة إلى غير الجيران تأخذ القيمة لا نهاية ∞.

Flash 4

إقلاع الجداول في التسيير المعتمد على شعاع المسافة

2.2.3 المشاركة Sharing

تكمن الفكرة الرئيسة من بروتوكول شعاع المسافة في مشاركة المعلومات المتوافرة لدى عقدة ما مع جيرانها. فإذا كانت العقدة A لا تعرف شيئاً عن العقدة E فإن العقدة C (جارة A) تعرف. فإذا استطاعت العقدة A مشاركة معلوماتها مع معلومات العقدة C فيمكن للعقدة A أن تصبح قادرة على معرفة كيفية الوصول إلى E.

تجري المشاركة فعلياً بالمعلومات الموجودة ضمن أول عمودين من جدول التسيير بينما يتم استبدال العمود الثالث (Next Hop) بعنوان العقدة المرسلة للجدول. أما فيما يتعلق بتواتر عملية الإرسال فهي إما أن تكون دورية أو عندما يحدث تغيير ما في الشبكة.

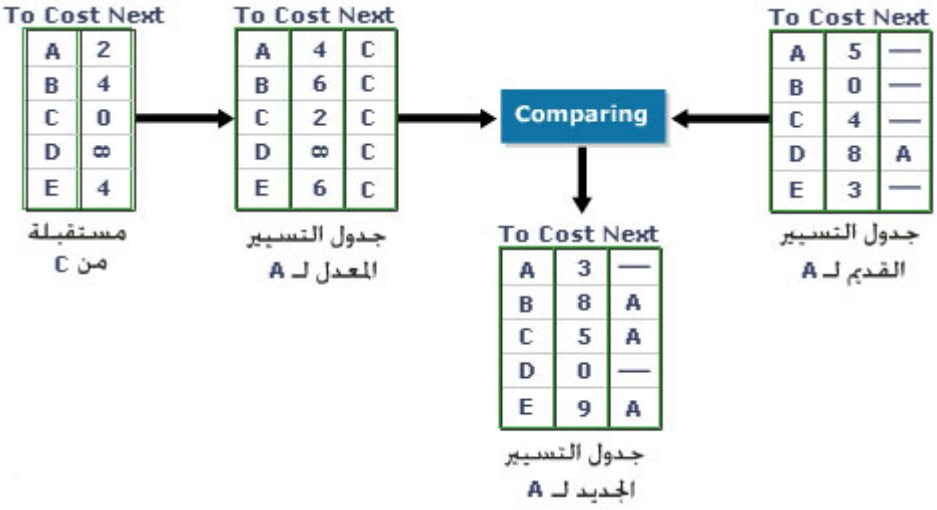
3.2.3 التحديث Updating

عندما تستقبل عقدة ما لجدول مكون من عمودين من جارة مباشرة لها فإنها تحتاج إلى تحديث جدولها بما يتناسب مع المعلومات المستقبلة. تتم عملية التحديث وفق ثلاث خطوات:

1. تحتاج العقدة المستقبلة إلى إضافة سعر الوصلة بينها وبين العقدة المرسلة إلى القيم الموجودة ضمن العمود الثاني Cost. إذا استقبلت العقدة رسالة من C التي تبعد عنها مسافة 2 وتزعم فيها أن مسافتها عن E هي 4 فتستطيع العقدة A أن تستنتج أن مسافتها عن E ستصبح 6 إذا أرادت المرور عن طريق C.
2. تحتاج العقدة المستقبلة إلى إضافة اسم العقدة المرسلة إلى عمودها الثالث من كل سطر في حال استخدمت المعلومات المستقبلة. أي أن العقدة المرسلة تصبح العقدة التالية في المسار.
3. تحتاج العقدة المستقبلة إلى مقارنة المعلومات الموجودة في كل سطر من جدولها القديم مع السطر المقابل ضمن النسخة المعدلة من الجدول المستقبل.
   - إذا كان مدخل العقدة التالية next hop entry مختلف فإن العقدة المستقبلة تختار السطر ذو السعر الأقل. في حال التعادل، تحتفظ العقدة بالقيمة السابقة.
   - إذا كان مدخل العقدة التالية هو نفسه، فإن العقدة المستقبلة تختار السطر الجديد.

لنفترض على سبيل المثال أن العقدة C كانت قد أعلنت مسبقاً أن سعر المسار إلى العقدة X هو 3. ولنفترض أنه لم يعد يوجد مساراً إلى العقدة X فإن العقدة C سوف تعلن أن سعر المسار إلى X قد أصبح ∞. يجب على العقدة A أن لا تتجاهل هذا السعر الجديد حتى ولو كان السعر الموجود ضمن جدولها القديم أقل وذلك لأن المسار القديم لم يعد له وجود.

يوضح الشكل التالي كيف تعدل العقدة A جدول تسييرها القديم بعد استقبال جدول العقدة C.



تحديث جدول التسيير في بروتوكول شعاع المسافة

لنناقش الآن السطر الأخير من الجدول المعدل، كان سعر المسار إلى E هو اللانهاية ضمن الجدول القديم لـ A لكن العقدة C أعلنت أنها تستطيع أن تصل إلى E بسعر 4 وبما أن سعر المسار من A إلى C هو 2 فهذا يعني أنه يمكن لـ A الوصول إلى E بسعر 2+4=6 عن طريق C.

لاحظ أيضاً أن المعلومات الموجودة ضمن الجداول تصبح بعد فترة بسيطة مستقرة ولا تتغير إلا في حال حدوث أعطال أو إضافات على الشبكة.

متى يجب المشاركة؟

السؤال الذي يطرح نفسه الآن هو متى يجب على عقدة ما إرسال الجدول الجزئي (لأنه يحوي عمودين فقط من أصل ثلاثة أعمدة) للتسيير إلى جميع جيرانها المباشرين؟

يجري عادةً إرسال الجدول إما بشكل دوري أو عندما تتغير الشبكة.

- **التحديث الدوري:**

ترسل كل عقدة جدول التسيير الجزئي إلى جميع جيرانها المباشرين بشكل دوري (عادةً كل 30 ثانية).

- **التحديث بالقدح Triggered update:**

ترسل عقدة ما جدول التسيير الجزئي إلى جيرانها المباشرين عندما يطرأ أي تغيير عليه.

يمكن أن ينتج التغيير عما يلي:

1. عندما تستقبل عقدة لجدول تسيير جزئي من عقدة أخرى الأمر الذي يؤدي إلى تغيير جدولها

2. تكتشف العقدة حدوث عطل ما أو انقطاع في الوصلات مع أحد جيرانها الأمر الذي يؤدي إلى زيادة سعر الوصلة المقطوعة إلى اللانهاية.

3.3 بروتوكول تسيير المعلومات RIP

يعتمد بروتوكول RIP على شعاع المسافة لتحقيق التسيير ضمن شبكات مترابطة مع إضافة بعض التعديلات، وهي:

1. نتعامل هنا مع مسيرات وشبكات (أو وصلات). توجد جداول التسيير ضمن المسيرات فقط.
2. الوجهة ضمن جدول التسيير هو شبكة أي أن العمود الأول يحدد شبكة وليس عقدة.
3. يتم تحديد السعر ضمن بروتوكول RIP على أنه عدد الشبكات الفاصلة بين مسيرين (لذلك تسمى عادةً عدد القفزات).
4. تعرف اللانهاية على أنها القيمة 16 أي أنه يجب أن لا يزيد طول أطول مسار ضمن الشبكة المترابطة عن 15 قفزة.
5. يعرف عمود القفزة التالية (next hop column) عنوان المسير الذي سيجري إرسال الطرد إليه للوصول إلى وجهته النهائية.

يبين الشكل التالي شبكة مترابطة تضم 7 شبكات و4 مسيرات وجدول التسيير ضمن كل مسير.

Img 7

مثال عن استخدام بروتوكول RIP

ملاحظة:

إن الأرقام ذات اللون الأحمر هي عنوان شبكة بكاملها بينما الأرقام ذات اللون الأسود فهي عنوان بوابة مسير.

لاحظ هنا أن جدول المسير R1 يحوي 7 مدخلات بمعدل مدخل لكل شبكة.

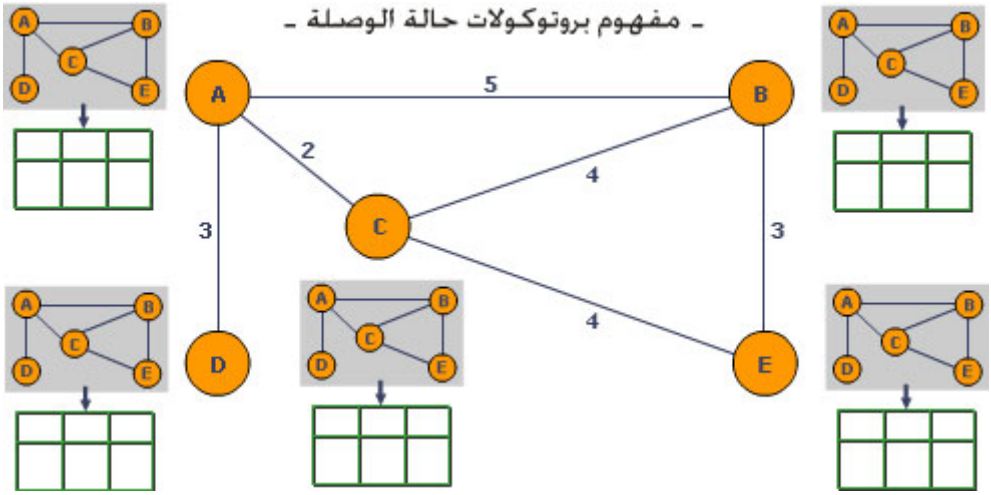
إن R1 موصول وصلاً مباشراً إلى الشبكتين: 130.10.0.0 و 130.11.0.0 الأمر الذي يعني أنه لا يوجد مدخل خاص بعنوان القفزة التالية.

حتى نستطيع إرسال طرد إلى الشبكات الثلاثة الواقعة في أقصى يسار الشكل، فإن R1 يحتاج إلى تسليم الطرد إلى R2 لذلك فإن عنوان القفزة التالية لهذه الشبكات الثلاثة هو عنوان بوابة المسير R2. لإرسال طرد إلى الشبكتين الواقعتين في أقصى اليمين فإن المسير R1 يحتاج إلى تسليم الطرد إلى المسير R4، لذلك وضع عنوان بوابة R4 وهـ و 130.11.0.1 ضمن جدول التسيير.

4.3 التسيير المعتمد على حالة الوصلة Link State Routing

تعتمد بروتوكولات حالة الوصلة على فلسفة مختلفة والتي تقول: "إذا كانت كل عقدة قادرة على معرفة الطبولوجية الكاملة لترابط الشبكات (أي خريطة العقد والوصلات وسعر كل وصلة وحالتها)، فإنها تستطيع استخدام خوارزمية Dijkstra لبناء جدول التسيير".

يوضح الشكل التالي المفهوم العام للبروتوكول.



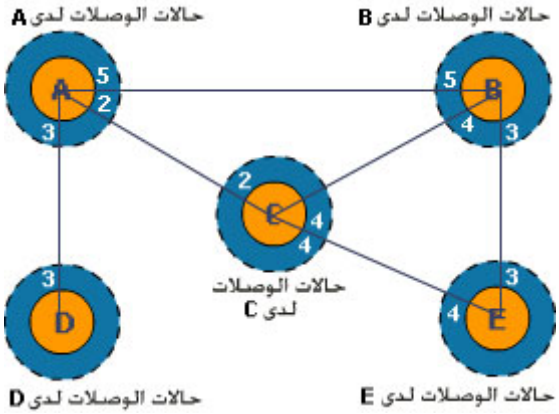
مفهوم برونوكولات حالة الوصلة

لاحظ هنا أن كل عقدة تمتلك خريطة ترابط الشبكات الكاملة لذلك فهي قادرة على حساب أقصر طريق للوصول إلى أي وجهة. لاحظ أيضاً أن وجود نفس الخريطة لدى جميع العقد لا يـؤدي إلى كون الطرق المحسوبة في كل عقدة متطابقة وذلك لأن كل عقدة تحسب أقصر الطرق إليها. الأمر مشابه لخرائط المدن التي يستخدمها السائحون، فرغم كون خريطة مدينة هي نفسها لدى جميع السائحين إلا أن المسار التي يسلكه سائح ما للوصول إلى منطقة يتعلق بنقطة الانطلاق.

السؤال الأهم هنا: كيف يمكن لكل عقدة أن تعرف الخريطة الكاملة للشبكة وأن تحدث هذه الخريطة في كل لحظة يطرأ تغيير عليها؟

لاحظ أيضاً أن كل عقدة تستطيع في أحسن الأحوال اكتشاف الوصلات التي تربطها بجيرانها فقط (أي لديها معلومات مجتزئة عن حالة الشبكة الكلية).

يوضح الشكل التالي هذه الفكرة مطبقة على نفس الشبكة السابقة.



معرفة حالات الوصلات

تعرف العقدة A أنها موصولة إلى كلٍ من B بسعر 5 وإلى C بسعر 2 وإلى D بسعر 3 وكذلك الأمر بالنسبة لبقية العقد.

1.4.3 بناء جداول التسيير

يتطلب التسيير المعتمد على حالات الوصلات أربع مجموعات من العمليات لضمان امتلاك كل عقدة لجدول تسيير يحوي أقصر الطرق (الطريق الأقل سعراً) إلى بقية العقد.

1. تنشأ كل عقدة طرد حالة الوصلة Link State Packet (LSP) الذي يحوي حالة وسعر وصلاتها مع جيرانها المباشرين.
2. تعمم هذا الطرد على جميع العقد (المسيرات) باستخدام خوارزمية تعويم Flooding فعالة وموثوقة.
3. تُشكل شجرة أقصر الطرق من قبل كل عقدة.
4. يحسب جدول التسيير على أساس شجرة أقصر الطرق المحسوبة مسبقاً.

إنشاء طرد LSP:

يحوي طرد LSP في الحالة العامة على كمية كبيرة من المعلومات لكننا سنقتصر على المعلومات الهامة منها مثل هوية العقدة وقائمة الوصلات ورقم تسلسلي وعمر. تفيد أول معلومتين في بناء الطبولوجية الكاملة للشبكة؛ يفيد الرقم التسلسلي في تسهيل عملية تعميم (أو تعويم) طرد LSP على جميع العقد؛ بينما يفيد عمر الطرد في منع بقاء الطرد لفترة طويلة ضمن الشبكة المترابطة.

يجري توليد طرود LSP في حالتين:

1. عندما يحدث تغيير في طبولوجية الشبكة المترابطة.

2. على أساس دوري. بما أنه لا توجد حاجة فعلية لإرسال طرد LSP دورياً إلا للتخلص من الطرود القديمة فيجري تطويل هذا الدور إلى حوالي 60 دقيقة أو حتى ساعتين.

تعويم طرود LSPs:

بعد أن تكون كل عقدة قد انتهت من تحضير طرد LSP فإنها تقوم بتعميمه على جميع العقد (وليس الجيران المباشرين). يجري هنا استخدام خوارزمية تعويم خاصة كما يلي:

1. ترسل العقدة، التي ولدت الطرد، نسخة عنه على كل بوابة من بواباتها.
2. عندما تستقبل عقدة ما لطرد LSP فإنها تقوم بمقارنته مع النسخة الموجودة لديها (أي الطرد السابق من نفس المصدر). فإذا كان الطرد المستقبل أقدم من الطرد الموجود مسبقاً (نختبر هنا الرقم التسلسلي للطرد) فإن العقدة المستقبلة تهمل الطرد. أما إذا كان أحدث فإنها تقوم بما يلي:
   - إهمال الطرد القديم والاحتفاظ بالطرد الجديد.
   - إرسال نسخة عن الطرد الجديد على جميع البوابات ما عدا البوابة التي وصل منها الطرد.

تشكيل شجرة أقصر الطرق باستخدام خوارزمية Dijkstra:

يتكون لدى كل عقدة، بعد استقبال طرود LSPs من جميع عقد الشبكة، خريطة الطبولوجية الكاملة لترابط الشبكات. لكن هذه الخريطة لا تنفع بشيء إذا لم نقم بحساب أقصر الطرق.

تتألف الشجرة من عقد ووصلات وجذر. يمكن الوصول إلى أي عقدة من عقد الشجرة باستخدام مسار واحد فقط.

تعريف: شجرة أقصر الطرق هي الشجرة التي يكون فيها كل طريق من الجذر إلى أي عقدة هو أقصر الطرق.

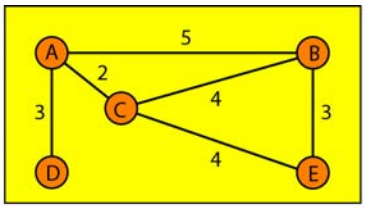
تُنشئ خوارزمية Dijkstra شجرة أقصر الطرق انطلاقاً من بيان Graph. تقسم الخوارزمية العقد إلى مجموعتين: مؤقتة Tentative ودائمة Permanent. تقوم بإيجاد جيران عقدة حالية وجعلهم مؤقتين واختبارهم حسب مواصفة ما لجعلها دائمة.

Img 10

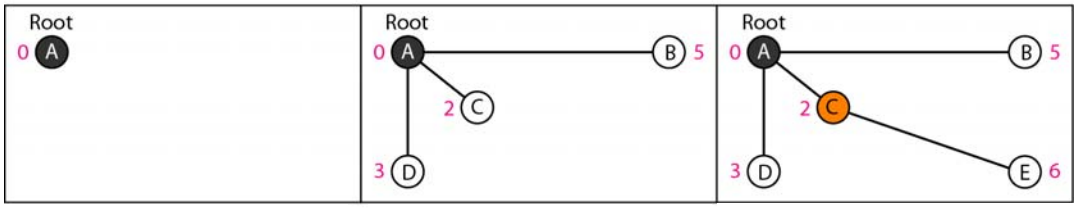
خوارزمية Dijkstra

لنقم بتطبيق الخوارزمية على العقدة A حسب ما هو موضح في الشكل التالي:

Img 11



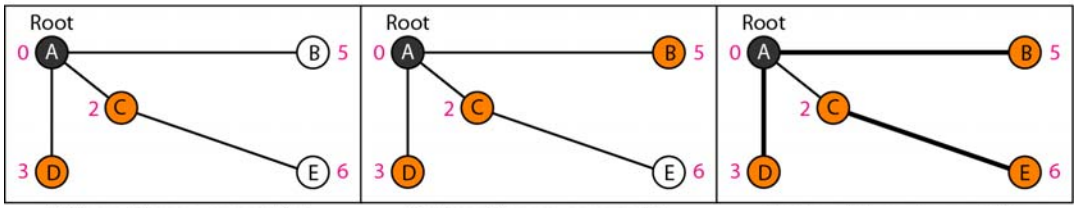
الطبولوجيا



1. اجعل A جذر root وانقله إلى القائمة المؤقتة

2. انقل A إلى القائمة الدائمة وأضف B و C و D إلى القائمة المؤقتة

3. انقل C إلى القائمة الدائمة وأضف E إلى القائمة المؤقتة



4. انقل D إلى القائمة الدائمة

5. انقل B إلى القائمة الدائمة

6. انقلُ إلى القائمة الدائمة (القائمة المؤقتة تصبح فارغة)

مثال عن تشكيل شجرة أقصر الطرق

1. نجعل العقدة A جذراً للشجرة وننقلها إلى القائمة المؤقتة (القائمة الدائمة فارغة والقائمة المؤقتة تحوي A(0))
2. تمتلك العقدة A أقل سعر تراكمي من بين جميع عقد القائمة المؤقتة. نقوم بإضافة A إلى القائمة الدائمة وإضافة جميع جيرانها المباشرين إلى القائمة المؤقتة (القائمة الدائمة تحوي A(0) والقائمة المؤقتة تحوي B(5), C(2), D(3)).
3. تمتلك العقدة C أقل سعر تراكمي من بين جميع عقد القائمة المؤقتة. ننقل C إلى القائمة الدائمة. يوجد للعقدة C ثلاثة جيران؛ لكن بما أننا قمنا مسبقاً بمعالجة العقدة A مما يجعل الجيران غير المعالجين هم B و E. لكن B موجود حالياً ضمن القائمـة المؤقتة بسعر تراكمي 5. تستطيع العقدة A الوصول إلى العقدة B عن طريق C لكن بسعر تراكمي 6. لكن بما أن القيمة القديمة 5 هي أقل فنقوم بالاحتفاظ بها ضمن القائمة المؤقتة (القائمة الدائمة تحوي A(0), C(2) والقائمة المؤقتة تحوي B(5), D(2), E(6)).
4. تمتلك العقدة D أقل سعراً تراكمي من بين جميع عقد القائمة المؤقتة. ننقل D إلى القائمة الدائمة. لا تحوي D أي جار لم يجر معالجته فلا نضيف أي عقدة إلى القائمة المؤقتة (القائمة الدائمة تحوي الآن A(0), C(2), D(3) والقائمة المؤقتة تحوي B(5), E(6)).

5. تمتلك العقدة B أقل سعراً تراكمي من بين جميع عقد القائمة المؤقتة. ننقل B إلى القائمة الدائمة. نحتاج الآن إلى إضافة جيران B الذين لم تجر معالجتهم بعد (أي العقدة E)؛ لكن بما أن السعر التراكمي للعقدة E السابق وهو 6 موجود داخل القائمة وهو أقل من السعر الجديد 8 فنحتفظ بالسعر القديم وبالعقدة E داخل القائمة المؤقتة. (القائمة الدائمة تحوي الآن A(0), B(5), C(2), D(3) والقائمة المؤقتة تحوي E(6)).
6. ننقل الآن العقدة E إلى القائمة الدائمة. بما أنها لا يوجد جيران لم تجر معالجتهم بالنسبة للعقدة E فتصبح القائمة المؤقتة فارغة ونوقف التنفيذ. تصبح شجرة أقصر الطرق جاهزة والقوائم النهائية هي: القائمة الدائمة تحوي A(0), B(5), C(2), D(3), E(6) والقائمة المؤقتة فارغة

حساب جدول التسيير من شجرة أقصر الطرق:

تستخدم كل عقدة بروتوكول شجرة أقصر الطرق لبناء جدول التسيير. يظهر جدول التسيير سعر الوصول إلى كل عقدة انطلاقاً من الجذر. الجدول التالي يظهر جدول التسيير الخاص بالعقدة A.

Node	*Cost*	*Next Router*
A	0	---
B	5	---
C	2	---
D	3	---
E	6	C

الجدول 1 - جدول تسيير العقدة A

إذا قارنت الجدول 1 مع الشكل تلاحظ أن التسيير المعتمد على شعاع المسافة والتسيير المعتمد على حالة الوصلة يوجدان نفس جدول التسيير.

5.3 البروتوكول المفتوح لإيجاد أقصر الطرق أولاً Open Shortest Path First (OSPF)

لزيادة فعالية بروتوكول OSPF، الذي يعتمد على حالات الوصلات، فإنه يقوم بتقسيم النظام المستقل إلى مناطق Areas. يجب أن تكون جميع الشبكات داخل منطقة واحدة مترابطين مع بعضهم البعض.

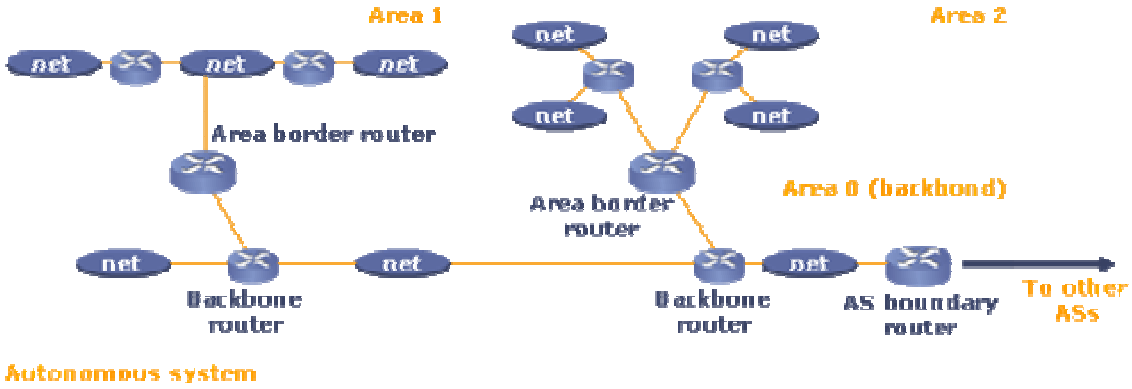
تعريف نظام مستقل Autonomous System (AS):

هو مجموعة من الشبكات والمسيرات تحت سلطة إدارة واحدة.

تقوم المسيرات داخل منطقة ما بتعويم معلومات التسيير داخل المنطقة فقط (تحوي المنطقة مسيرات ومحطات وشبكات). يجري وضع، على حدود كل منطقة، مسيرات خاصة تدعى **مسيرات حدود المنطقـة** تسند إليها مهمة تلخيص معلومات المنطقة وإرسالها إلى المناطق الأخرى.

يجري أيضاً تعريف منطقة خاصة تدعى **المنطقة الفقارية Area Backbone** وتدعى المسيرات الموجودة داخلها بالمسيرات الفقارية أيضاً. يجب أن تكون جميع المناطق موصولة إلى المنطقة الفقارية.

تخصص كل منطقة **بمعرف منطقة Area Identification** وتكون قيمة معرف المنطقة الفقارية صفر. يظهر الشكل التالي نظام مستقل والمناطق المستخدمة.



تشكيل المناطق ضمن نظام مستقل

المقياس Metric:

يسمح بروتوكول OSPF لمدير النظام بتحديد سعر، يدعى Metric، لكل طريق. يمكن أن يكون المقياس متعلقاً بجودة الخدمة كزمن التأخير أو معدل نقل المعطيات الأعظمي وغيرها.

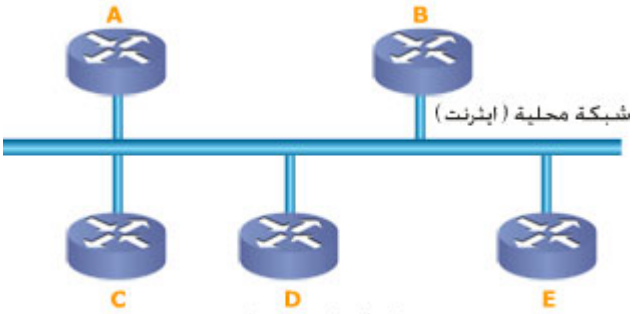
نوع الوصلة: يدعى الاتصال وفق بروتوكول OSPF باسم وصلة. توجد أربعة أنواع للوصلات:

- نقطة لنقطة Point-to-point
- عابرة Transient
- عقب Stub
- افتراضية Virtual.

تربط **وصلة نقطة لنقطة** مسيرين فقط دون وجود أي مسير أو محطة بينهما. يجري أيضاً تحديد المقياس الذي يكون عادةً نفسه على طرفي الوصلة كما هو مبين في الشكل التالي.

- شبكة نقطة لنقطة -

الوصلة العابرة: هي عبارة عن شبكة موصول إليها عدة مسيرات. يمكن أن تدخل المعطيات عن طريق أي مسير وتخرج عن طريق أي مسير آخر. تنتمي جميع الشبكات المحلية وبعض الشبكات الواسعة إلى هذا النوع من الوصلات.



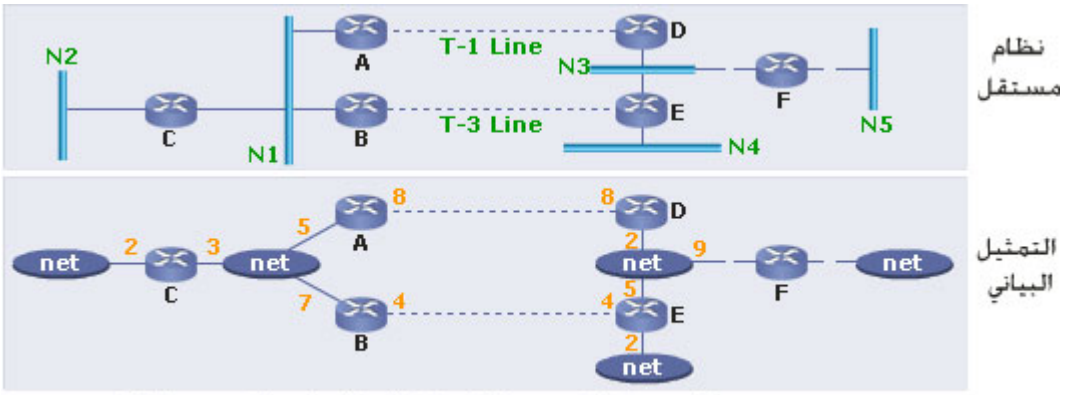
- الشبكة العابرة -

الوصلة العقب: هي عبارة عن شبكة موصولة إلى مسير واحد فقط.

- الشبكة العقب -

يستطيع مدير النظام، عندما يتم انقطاع الوصلة بين مسيرين، إنشاء **وصلة افتراضية** بينهما.

التمثيل البياني: يبين الشكل التالي التمثيل البياني لنظام مستقل AS مؤلف من سبع شبكات وستة مسيرات



- مثال عن نظام مستقل و تمثيله البياني وفق بروتوكول -

نستخدم هنا الرموز N1, N2 للشبكات العابرة وشبكات العقب والمستطيلات للدلالة على المسيرات والأشكال البيضوية للدلالة على الشبكات. تدل الرموز T-1 و T-3 على أنواع من الدارات المخصصة نقطة لنقطة التي يجري عادةً استئجارها من مزود خدمة الاتصالات.