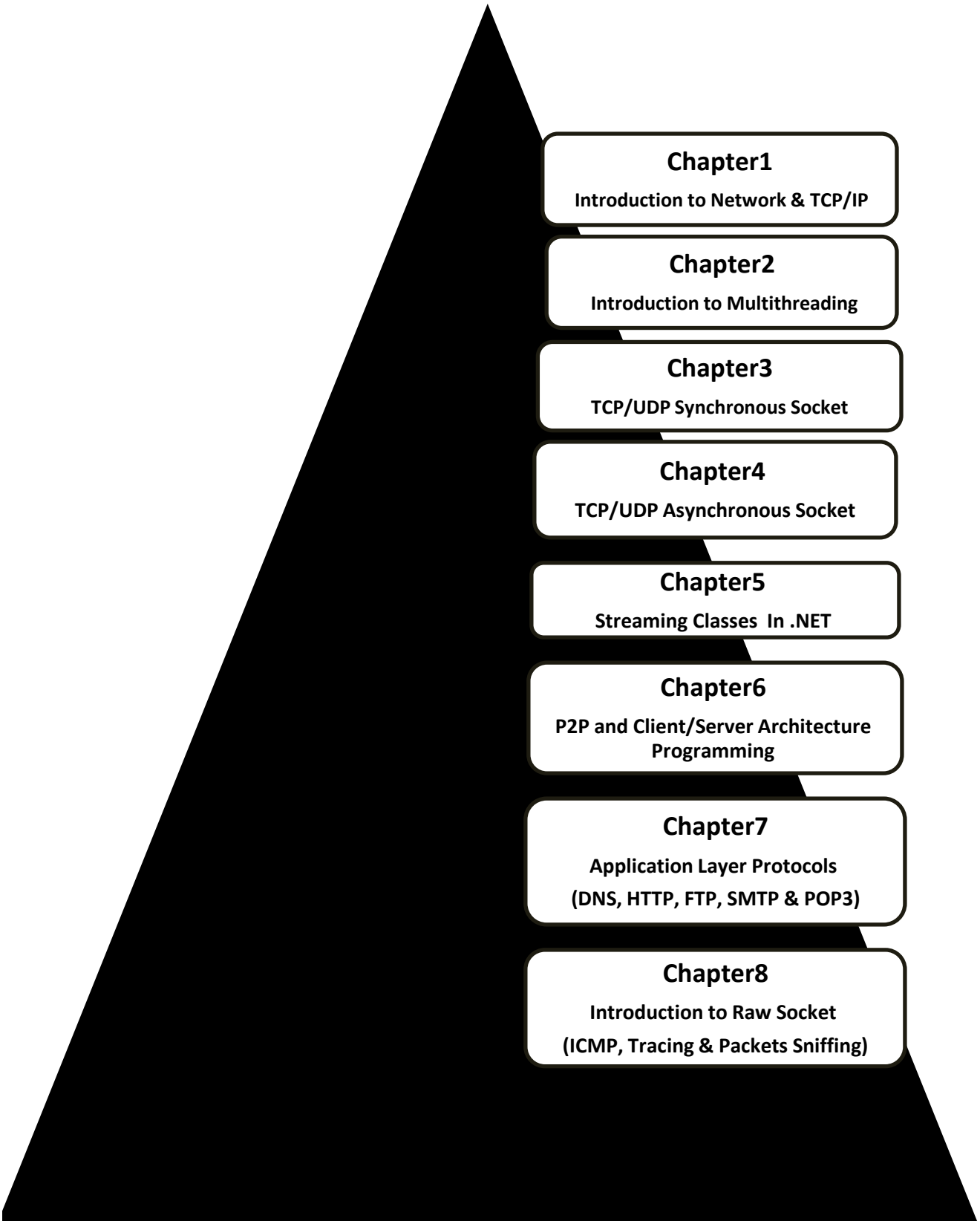
PAERT 1

Basic Network Programming

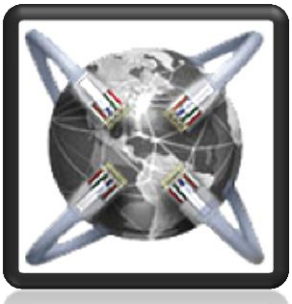


Chapter 1

Introduction to TCP/IP Networking

المواضيع الرئيسية في هذا الفصل:

1- مقدمة في بروتوكول TCP/IP وطرق الاتصال والتراسل.
2- مقدمة في مبدأ عمل بروتوكولات طبقة الـApplication Layer
3- مقدمة في مبدأ عمل بروتوكولات طبقة الـTransport Layer
4- مقدمة في مبدأ عمل بروتوكولات طبقة الـNetwork Layer



1 - مقدمة الفصل الأول:

نقوم في حياتنا اليومية بأداء آلاف الأمور والتي تستخدم برمجيات الشبكات وسواء قمنا بإرسال الملفات أو تبال البريد الإلكتروني أو الدردشة عبر الإنترنت فإن أول ما نقوم به هو تحضير البيانات باستخدام التطبيقات المخصصة لأداء تلك الوظائف ومن ثم إرسال تلك البيانات وسواء كنت تتصل مع الشبكة بالطريقة السلكية أو ألا سلكية فإن الوظائف التي تستخدمها لعملية الإرسال هي نفسها ولن تحتاج إلى إجراء تعديلات في إعداد البرنامج الذي تستخدمه والسبب الرئيسي لذلك هو انك تعمل على بروتوكول يقسم عملية الاتصال إلى مجموعة من المراحل وهذا البروتوكول هو الـ TCP/IP وفي العادة فإن أغلب عمل المستخدم لن يتعدى طبقة الـApplication Layer وفي هذه الطبقة لن تضطر إلى التعامل مع كيفية إجراء الاتصال أو الإعدادات المتعلقة به وكل ما تقوم به هو تحضير البيانات ومن ثم إرسالها ومن هنا نعرف أن أي عملية اتصال لابد أن تمر بمجموعة من المراحل وقد قسمت منظمة الـ IOS – International Organization For Standardization بالاتفاق مع الـITU-International Telecommunication Union عملية الاتصال إلى سبعة مراحل تسمى OSI - Open Systems Interconnection Model وهذا يعني آن أي عملية اتصال سواء باستخدام الـTCP/IP أو IPX/SPX أو أي Standard أخر فإنه يجب أن يمر بهذه المراحل. وبأبسط صورة فإن هذه المراحل هي:

1- مرحلة تحضير البيانات باستخدام واجهة المستخدم الـ User Interface باستخدام إحدى التطبيقات الـApplication وسنأتي على شرح هذه المرحلة بالتفصيل في هذا الفصل والفصول اللاحقة.
2- مرحلة تمثيل البيانات إذ انه ومن الطبيعي أن يتم تحويل البيانات إلى الشكل الذي يجعله قابلا للحفظ أو للإرسال وفي هذا الجزء في التحديد فنحن نتكلم عن تمثيل البيانات Data Presenting من Data إلى Binary ففي برمجيات الدردشة Chatting Application يتم تمثيل البيانات المراد إرسالها من نص Text إلى مجموعة من الـBits وتسمى هذه المرحلة مرحلة الـEncoding ومن أنواعه الـASCII (American Standard Code for Information Interchange) والذي يستخدم 1Byte أي 8 Bits لتمثيل حرف واحد في اللغة الانجليزية والبت هو تعبير عن قيمة 0 أو 1 ومن الأنواع الأخرى للـEncoding الـUnicode أو الـUniversal Code ويستخدم من 16-32 Bits لتمثيل حرف واحد لمعظم لغات العالم – راجع الـEncoding Part في الملحقات لمزيد من المعلومات.
3- مرحلة تحضير الاتصال قبل البدا بعملية الإرسال ويسمى (إنشاء الجلسة) Session Establishing وفيها البدء بعملية التخاطب بين الجهازين والتعريف بخصائص الاتصال وتتم عادة في الـTCP بمجموعة من المراحل تسمى Three Way Hand Shake أما في الـUDP فيتم البدء بالإرسال دون إجراء أي من عمليات التحقق السابقة ويوجد نوعين من أنواع الـSessions الاتصال المتبادل Full-

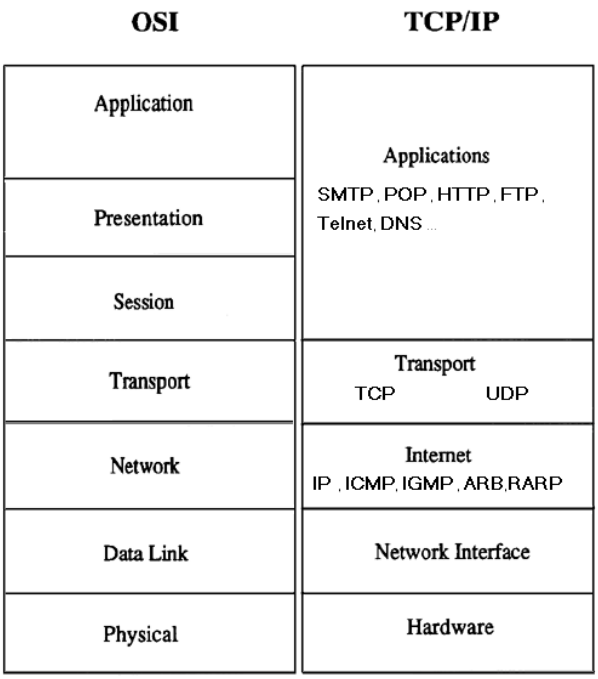
Duplex ويعني إمكانية الإرسال والاستقبال في نفس الوقت والاتصال أحادي الاتجاه Half-Duplex ويعني إما الإرسال أو الاستقبال في نفس الوقت وسنأتي على شرح كل منها بتفصيل في هذا الفصل والفصول اللاحقة.

4- مرحلة اختيار بروتوكول النقل Transport Protocol ففي الـTCP/IP فإنه يوجد ثلاثة بروتوكولات نقل رئيسية وهي الـ UDP-User Datagram Protocol والـ TCP – Transmission Control Protocol والـSCTP – Stream Control Transmission Protocol آما الأول والثاني فيستخدم بشكل أساسي في نظام التشغيل Windows و Linux وأما الأخير فهو مدعم فقط في النظام Linux ويتم أيضا في هذه المرحلة عملية العنونة الـSource & Destination Port Number على مستوى الـApplication ويستخدم الـPort Number لمعرفة التطبيق الذي ينقل أو يستقبل البيانات، وكمثال على ذلك بروتوكول الـHTTP والذي يستخدم الـPort 80 ويعتمد على الـTCP في عملية النقل لمزيد من المعلومان انظر الجدول 1-1 وسنأتي على شرح هذه المرحلة بالتفصيل في هذا الفصل والفصول اللاحقة.

5- العنونة ويتم فيها تحديد عنوان الجهة المستقبلة ووضع عنوان الجهة المرسلة للبيانات Source & Destination IP Address وسنأتي على شرح هذه المرحلة بتفصيل في هذا الفصل و الفصول اللاحقة.

6- التحضير لإرسال البيانات عبر وسيط النقل ويتم فيه استخدام العنوان الفيزيائي Physical Address للجهة المرسلة والمستقبلة وتحديد طريقة النقل التي ستتم على الوسط الفيزيائي ومن المعايير المستخدمة في هذه الطبقة PPP-Point to Point Protocol والـEthernet وغيره وسنأتي على شرح هذه المرحلة بالتفصيل في هذا الفصل و الفصول اللاحقة.

7- الإرسال عبر وسيط النقل الفيزيائي ومنه الاتصال السلكي IEEE 802.3 أو ألا سلكي IEEE 802.11 وغيره وسنأتي على شرحه بالتفصيل في هذا الفصل و الفصول اللاحقة.



الشكل 1-1 ويبين الـ TCP/IP والـOSI Stack

وكما هو مبين في الشكل 1-1 فإن عملية الاتصال باستخدام الـTCP/IP تمر بمراحل الـOSI كاملة غير انه تم اختصار الثلاثة طبقات الأولى في طبقة الـApplication والسبب الرئيسي وراء ذلك أن بروتوكولات الـApplication Layer في TCP/IP مثل HTTP, FTP, SMTP, SNMP, DNS, Telnet, POP وغيرها تقوم بهذه المراحل الثلاث الأولى واعتبرت كجزء من بنية البروتوكول نفسه.

1.1 إنشاء الاتصال وإرسال واستقبال البيانات من خلال بروتوكول الـTCP/IP:

تمر عملية إنشاء الاتصال ومن ثم إرسال واستقبال البيانات بعدة مراحل ففي الـApplication Layer يقوم المستخدم بتحضير البيانات وفي الـPresentation Layer تتم عملية الـEncoding بتمثيل البيانات إلى الشكل الذي يمكن من خلاله فهم الرسالة المرسلة والمستقبلة و في الـ Session Layer يقوم البرنامج بإنشاء الاتصال بين الطرفين المرسل والمستقبل والتحقق من تزامن الاتصال وتتم هذه المرحلة بعدة خطوات وهي:

1- إجراء الاتصال المبدئي بالـ Server عبر الـ IP والـ Port المحدد وذلك بعد تحديد عملية الاتصال سواء عبر الـ UDP أو عبر الـ TCP ومن المعروف أن جميع بروتوكولات الـApplication Layer في الـTCP/IP قد تم بنائها وتعريفها مسبقا بالبروتوكول الذي سيتم استخدامه في الـTransport Layer وكما هو واضح في الجدول 1-1.

2- عمل الـ Authentication المطلوب إذا تطلب الـServer ذلك.

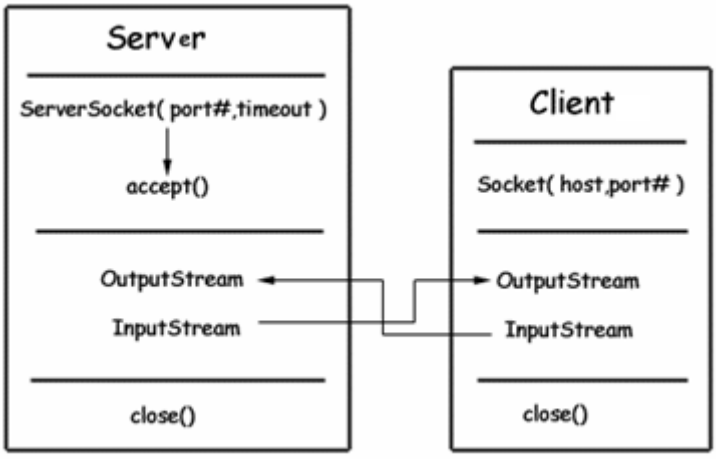
3- قبول أو رفض الـSession ويتم ذلك بإرسال الموافقة على إنشاء الـSession أو رفضها

4- إنشاء الـSession وقيام الـServer بعمل Listening على الـPort الخاص بالبرنامج

TCP/IP Application Layer Protocol	Transport Protocol الافتراضي	Listener Port Number الافتراضي
HTTP , HTTPS Hypertext Transfer Protocol	TCP	80 443
FTP File Transfer Protocol	TCP	20 21
SMTP Simple Mail Transfer Protocol	TCP	25
SNMP Simple Network Management Protocol	TCP	161 162
Telnet	TCP	23
POP3 Post Office Protocol Version 3	TCP	110
DNS Domain Name System	UDP	53 137

الجدول 1-1 لمزيد من المعلومات يمكن الرجوع إلى رابط منظمة الـIANA http://www.iana.org/assignments/port-numbers

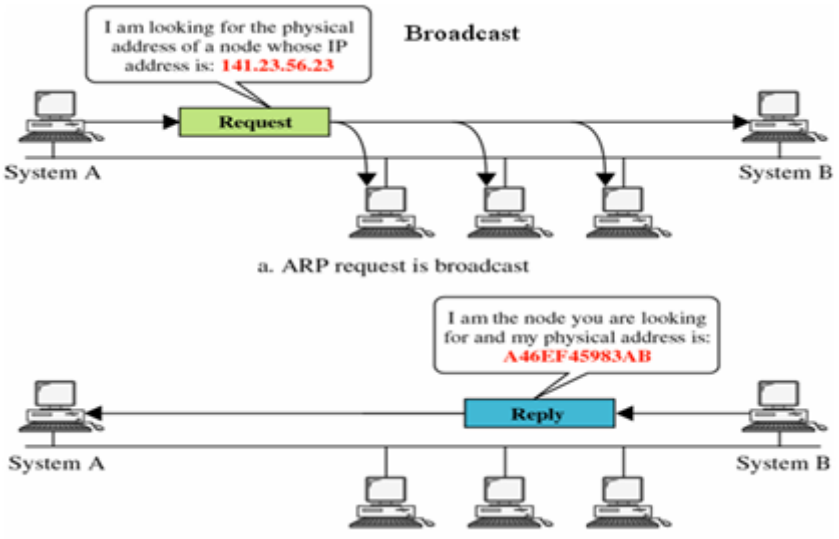
عندما يتم الموافقة على إنشاء الـSession والبدء بعملية التخاطب يقوم جهاز المرسل الـ Client بتحميل الرسالة إلى الطبقة الأخرى وهي طبقة الـ Transport وفي هذه الطبقة يتم تحديد طبيعة الاتصال سواء عبر TCP - Connection Protocol أو عبر الـUDP-Connectionless Protocol ففي البروتوكول الأول يتم تحديد طرفين وهما المرسل والمستقبل وPort الاتصال أما باستخدام الـ UDP فيمكن أن يكون الطرف المستقبل كل الأجهزة Broadcast أي لا وجود لعنوان محدد يستقبل الرسالة وهذا يعني أن أي نظام يقوم بتصنت عبر هذا الـPort يستطيع استقبال الرسالة ، كما يمكن من عمل الـMulticasting (الإرسال إلى مجموعة Group)، ويتم كل ذلك بوضع الـUnicast IP أو الـBroadcast IP أو الـMulticast IP مع رقم الـPort في وحدة واحدة وتسمى الـSocket فالـSocket هي عملية ربط بين طبقتين في الـTCP/IP وهي طبقة الـTransport والـNetwork Layer ويوضح الشكل 1.2 مبدأ عمل الـSocket.



الشكل 1.2 ويبين مبدأ عمل الـSocket في الـTCP/IP Stack

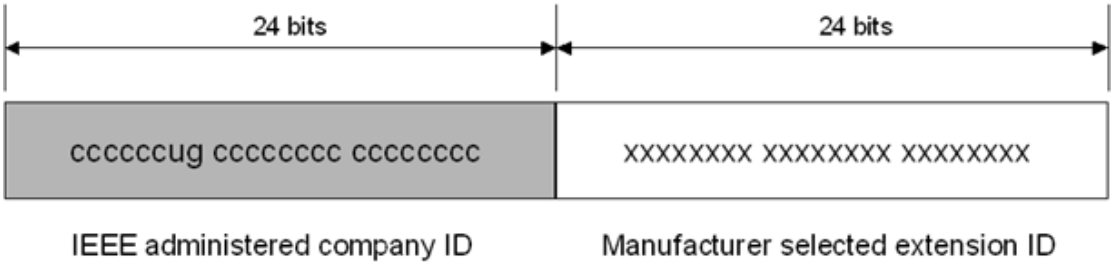
وبعد ذلك تسلم البيانات إلى الـ Network Layer إذ تتم عنونة الرسالة ووضع عنوان المرسل والمستقبل وتسلم إلى الطبقة الأدنى ليتم إرسالها عبر الـ Physical Tunnel ويحتوي الـNetwork Layer على مجموعة من البروتوكولات منها IP,IPv6,ARP-Address Resolution Protocol وغيرها (حيث أن الـ Network Layer في TCP/IP تعبر عن الـ Network Layer + Data Link Layer In OSI Model) ولكل منها وظيفة معينة سنأتي على شرحها بالتفصيل في هذا الفصل والفصول اللاحقة وتسمى البيانات في هذه المرحلة بالـPacket.

بعد إجراء عملية العنونة يقوم المرسل بسؤال عن عنوان الـPhysical Address الخاص بالـDestination ويسمى الـMAC Address وتسمى البيانات في هذه المرحلة بالـFrame وتتم هذه العملية عبر بروتوكول الـARP-Address Resolution Protocol ويقوم هذا البروتوكول بالتحقق من وجود الـMAC Address في الـARP Table وفي حالة عدم وجوده فإنه يرسل ARP Request Broadcast Message إلى كل الأجهزة على الشبكة يسأل فيها عن صاحب الـIP Address المراد الإرسال له فإذا وجده يرسل الجهاز المعني Unicast ARP Replay Message يخبره فيها بعنوان الـMAC Address الخاص به وبعد استلام الرسالة يقوم بتخزين العنوان المعني في الـ ARP Table لاستخدامه في المرات اللاحقة وتتضح هذه المرحلة في الشكل 1.3 ومن المعروف أن أي Network Interface يحتوي على عنوان مصنعي وهو عنوان فريد Unique على مستوى العالم ويسمى الـMAC- Media Access Control address ويكتب في العادة بالـHexadecimal ويتكون من 48 Bits وكما هو واضح في الشكل 1.4 فإن الـMAC Address يقسم إلى جزأين 24 Bits تعبر عن الرمز الخاص بالمصنع وهي ثابتة تمنح من قبل اتحاد الـIEEE و 24 Bits أخرى يتم وضعها من قبل المصنع نفسه كرقم تسلسلي فريد لكل منتج من منتجاته.

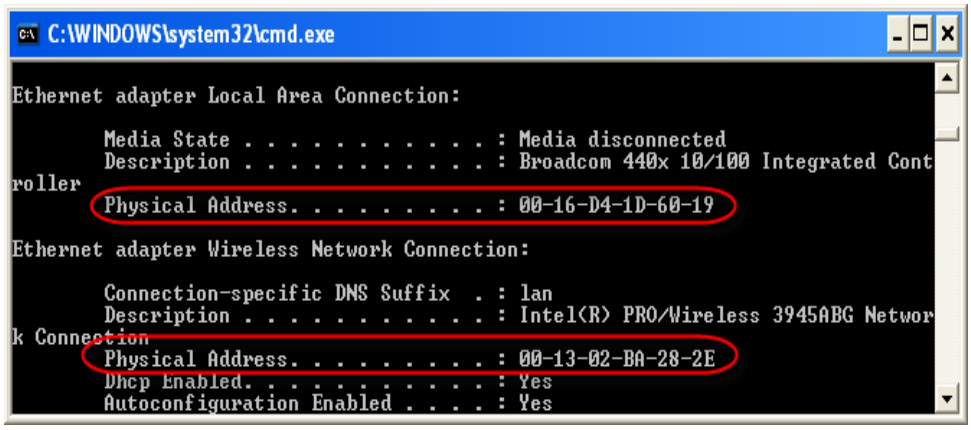


الشكل 1.3 ويبين كيفية عمل الـARP في شبكات الـEthernet

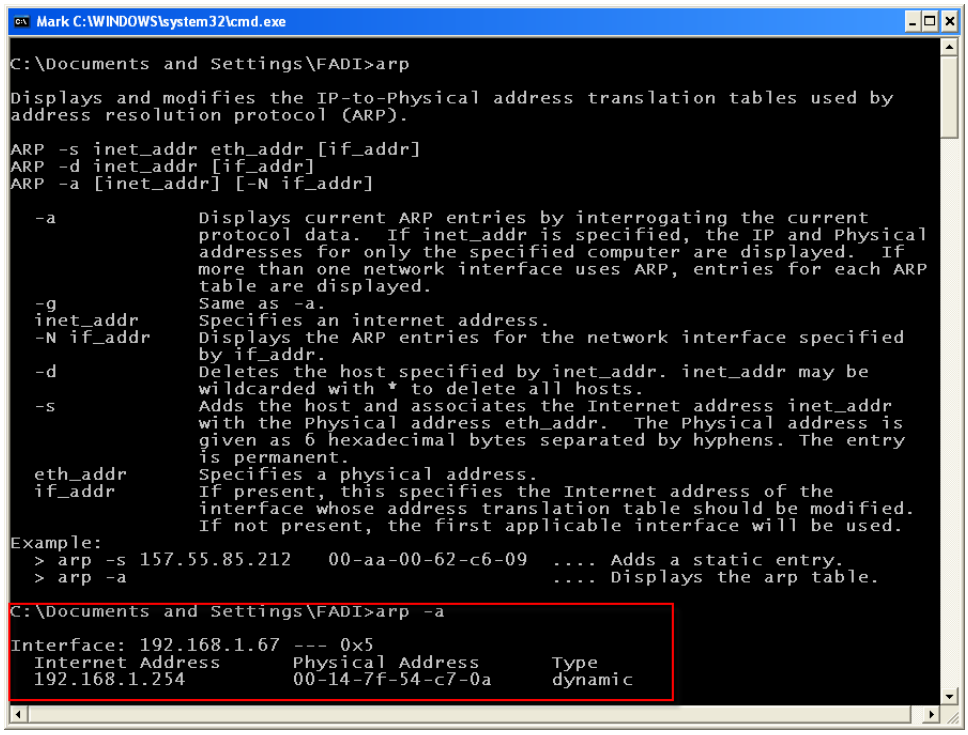
ويستخدم العنوان FF:FF:FF:FF:FF:FF كـ Broadcast MAC Address عند قيام الـARP بالاستفسار عن عنوان الـMAC لجهاز ما على الشبكة إذ يوضع ذلك العنوان في الـDestination Address ضمن الـ ARP Protocol وكما هو واضح في الشكل 1.5 يمكن معرفة الـMAC Address لجميع الـInterfaces في نظام التشغيل ويندوز بكتابة الأمر ipconfig/all في الـCommand prompt وكما يمكن الوصول إلى الـ ARP Table بكتابة الأمر arp –a وكما هو واضح في الشكل 1.6.



الشكل 1.4 ويبين تقسيم الـMAC Address



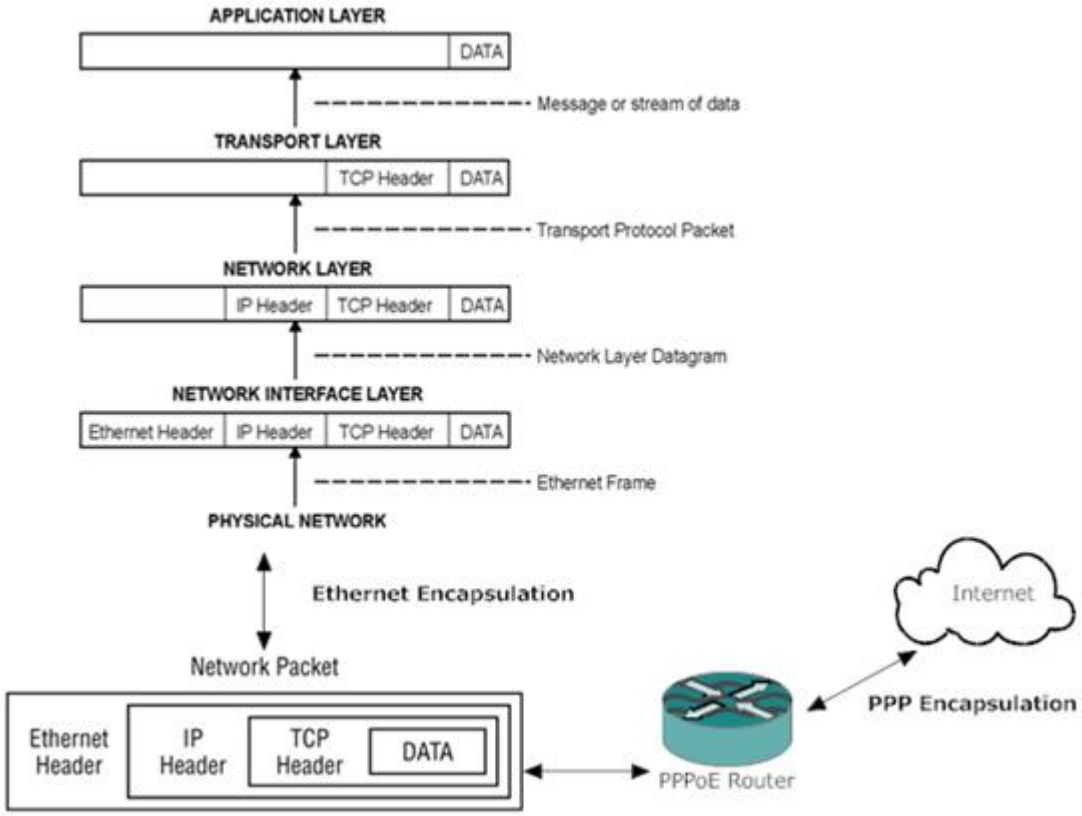
الشكل 1.5 ويبين كيفية الوصول إلى الـMAC Address



الشكل 1.6 ويبين كيفية الوصول إلى الـARP Table

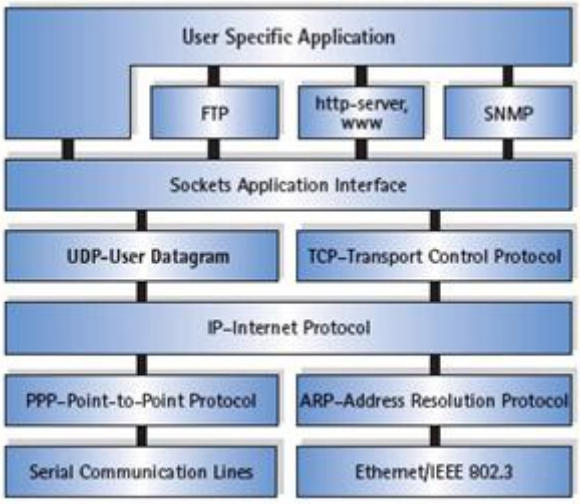
وبعد مرحلة جلب الـMAC يتم تحديد نوع الـEncapsulation هل سيكون كـSerial Encapsulation أو Ethernet Encapsulation ويتم في الـData Link Layer وفي حالة كنت تتصل بالإنترنت من خلال Router فإن الـEncapsulation سيكون في الأغلب Ethernet بينك وبين الـRouter ومن ثم تتم عملية إرسال الـFrame من الـRouter إلى الـGateway كـSerial Encapsulation وكما هو موضح في الشكل 1.7 ومن أنواعه الـPPP – Point to Point Protocol وأكثر طرق الاتصال بالإنترنت

شيوعا هي PPPoE – Point to Point over Ethernet وكذلك PPPoA – Point to Point Over ATM و أيضا الـFrame Relay.



الشكل 1.7 ويبين عملية الـEncapsulation للبيانات كـPPPoE بين الشبكة الداخلية والـRouter

ويجب التفريق بين عملية الإرسال باستخدام الـEthernet وعملية الإرسال باستخدام Modem داخلي تتصل منه مباشرة بالـInternet إذ أنه في الـEthernet تتم عملية الإرسال بعد معرفة الـMAC Address لطرف الأخر باستخدام الـARP أما الـInternal Modem فيتم الوصول وفق مبدأ الـ PPP – Point to Point Protocol مباشرة لاحظ الشكل 1.8.

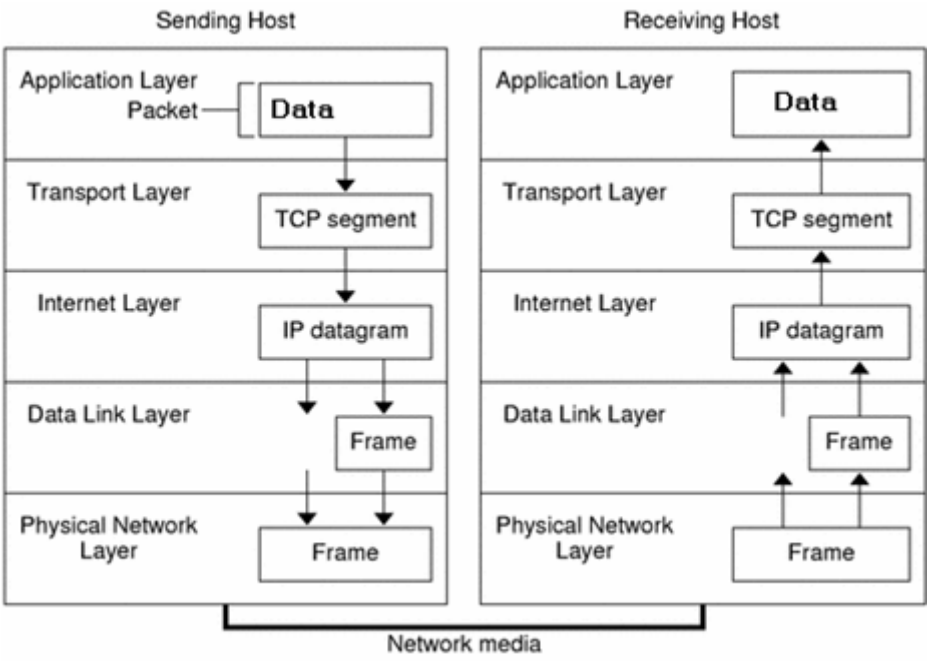


الشكل 1.8 ويبين الفرق في الـEncapsulation بين استخدام الـEthernet واستخدام الـModem مباشرة

في كل مرحلة من مراحل الـTCP/IP يتم إضافة Header على الـData يحتوي على معلومات التحكم لتلك المرحلة، ويتراوح حجم الـHeader في الـ Transport والـ Network Layer من 20 إلى 60 بايت حسب الـOptions التي يتم إضافتها إلى الـ Header ويعتبر حجم الـFrame من الأمور الهامة جدا في برمجة الشبكات وخاصة في حالة الإرسال عبر بروتوكول لا يدعم عملية تجزئة الـData مثل الـUDP إذ لا يمكن إرسال البيانات إلا كدفعة واحدة وبذلك لن تستطيع إرسال البيانات عبر الـUDP إذا زادت عن الحجم الأقصى للـEthernet Frame وهو 1500 Bytes وهي مقسمة كما يلي:

"The minimum size of an Ethernet frame is 64 bytes. The breakup of this size between the fields is: Destination Address (6 bytes) + Source Address (6 bytes) + Frame Type (2 bytes) + Data (46 bytes) + CRC Checksum (4 bytes). The minimum number of bytes passed as data in a frame must be 46 bytes. If the size of the data to be passed is less than this, then padding bytes are added. The maximum size of an Ethernet frame is 1518 bytes. The breakup of this size between the fields is: Destination Address (6 bytes) + Source Address (6 bytes) + Frame Type (2 bytes) + Data (1500 bytes) + CRC Checksum (4 bytes). The maximum number of bytes of data that can be passed in a single frame is 1500 bytes" ref http://rfc.net/rfc1705.html

وكما هو واضح في الشكل 1.7 يتم الاستقبال في الطرف الآخر بنفس المراحل ولكن بالعكس حيث يستلم الـ Network Adapter Card الـFrame من وسيط النقل الفيزيائي لتدخل بعد ذلك في مرحلة الـ Data link Layer ثم إلى الـ Network Layer ثم الـTransport Layer ثم الـApplication Layer وباستخدام الـ Encoding في الـPresentation Layer تحول البيانات من الـBits إلى نفس الصورة التي كانت عليها قبل الإرسال وتعرض على شاشة البرنامج الـApplication.

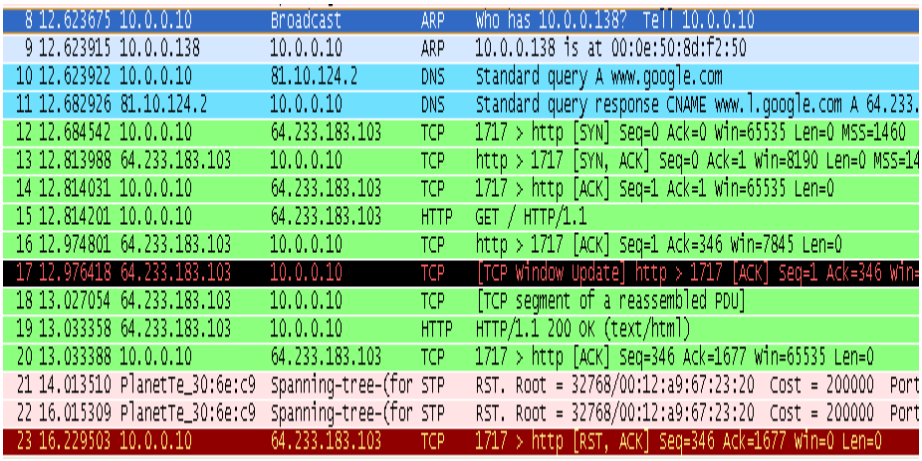


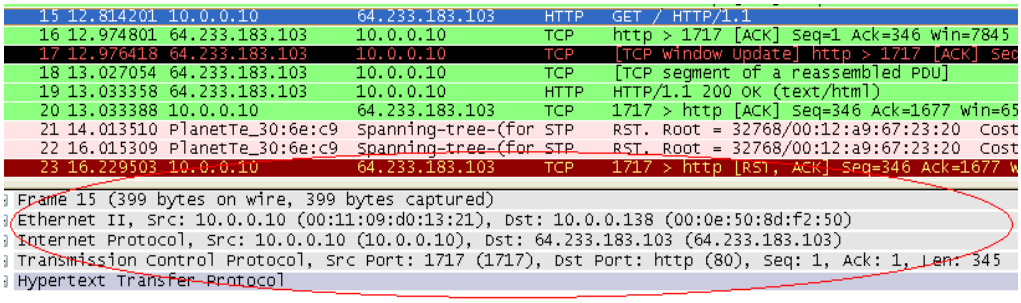
الشكل 1.9 ويبين مراحل إرسال واستقبال البيانات في الـTCP/IP

في المثال التالي سنقوم باستخدام برنامج الـEthereal وهو برنامج يقوم بعملية الـ Packet Sniffing والذي يمكن تحميله من الموقع الخاص به وهو: http://www.ethereal.com/download.html

مثال على استخدام الـARP والـDNS والـHTTP من خلال Router يستخدم الـNAT لمشاركة Public IP:

لاحظ الشكل 1.10 ويبين صورة مأخوذة من برنامج الـEthereal والذي يقوم بعمليات التصنت والتحليل للـFrames المارة من خلال الـNetwork Interface Card ، حيث قمنا بطلب الدخول إلى موقع الـGoogle باستخدام الـInternet Explorer بحيث أننا نتصل بالإنترنت من خلال ADSL Router عبر PPPoE.



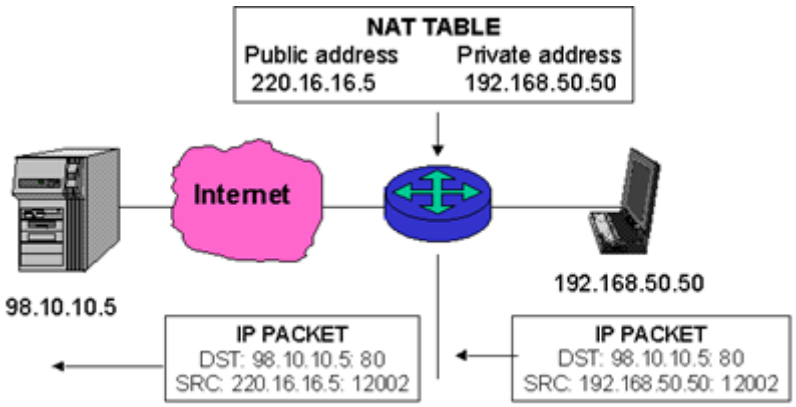


الشكل 1.10 ويبين تحليل للـFrames والتي استخدمت للوصول إلى الإنترنت لجلب موقع Google عبر الـHTTP

لاحظ أن أول عملية كانت السؤال عن الـMAC Address الخاص بالـGateway وهو هنا 10.0.0.138 حيث أرسلت هذه الرسالة كـBroadcast Message وفي العملية الثانية قام الـRouter بالرد على الـARP Request وأرسل عنوان الـMAC Address الخاص به بعد هذه العملية سيقوم الـClient بالاستفسار عن عنوان الـGoogle بإرسال DNS Request Query ، لاحظ أنه في المثال قد وجد الـ IP Address الخاص بالـGoogle في الـDNS Table والموجودة ضمن نظام التشغيل وهذا يعني انه تم الدخول على الـGoogle من قبل والدليل على ذلك أن الـIP Address الخاص بالـGoogle موجود في الـDNS Table وبعد هذه العملية قام موقع الـGoogle بالرد على الطلب يخبره بوجوده وضمن العنوان المحدد لاحظ أن العمليات من 8 إلى 20 هي عمليات طلب و إرسال لمحتويات الصفحة الخاصة بموقع الـGoogle وتتم باستخدام الـ HTTP Protocol.

ويتبين لنا من المثال السابق أن عملية استخدام الـHTTP لدخول إلى موقع عبر الإنترنت مرت بهذه المراحل:

1- يقوم المستخدم بطلب الدخول إلى موقع على الإنترنت وكمثال http://www.google.com يتم تحويل ذلك الطلب من خلال بروتوكول الـHTTP إلى الـDNS لمعرفة الـIP المقابل لهذا العنوان.
2- يتم البحث في الـDNS Table الموجود في نظام التشغيل وفي حالة لم يجده يقوم بطلب الـIP من الـDNS Server والذي يزودك به في العادة الـISP – Internet Service Provider الخاص بك.
3- يرسل طلب الـDNS إلى الـGateway ويعتبر في مثالنا الـRouter الداخلي كـGateway لشبكة الداخلية.
4- يتم البحث عن الـMAC Address الخاص بالـGateway الداخلي من خلال الـARP.
5- بعد جلب الـMAC ينتظر الـApplication الحصول على الـIP الخاص بالموقع وبعد ذلك يتم عمل HTTP Request للموقع إلى الـGateway الداخلي ويقوم نظام التشغيل بتوليد Port فريد على مستوى الـApplications يوضع في الـSource Port للـPacket ويتم إرساله إلى الـGateway المقابل المعرف في الشبكة الداخلية إذ ترسل من الحاسب إلى الـGateway كـEthernet Encapsulation ثم من الـRouter إلى الـGateway المقابل في الإنترنت كـPPP Encapsulation.
6- في المثال السابق فنحن نستخدم الـNAT – Network Address Translation لدخول إلى الإنترنت وهذا يعني أن الحاسب الداخلي يستخدم Private IP ويتشارك مع مجموعة الحواسيب الأخرى التي على نفس الشبكة بالـPublic IP والذي يتصل من خلاله الـRouter بالـGateway المقابل كما هو مبين في الشكل 1.11.



الشكل 1.11 ويبين اتصال الجهاز الداخلي بالشبكة الإنترنت من خلال مشاركة الـIP باستخدام الـNAT

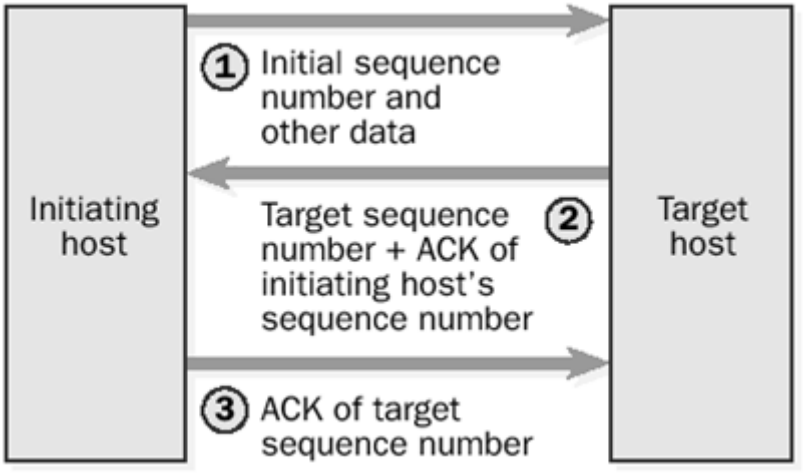
ويبين الشكل 1.11 كيف يقوم الـRouter بمشاركة الـPublic IP لكل الحواسيب الموجودة في الشبكة الداخلية حيث قام الـClient والذي يملك الـIP 192.168.50.50 بإرسال الطلب إلى الـGateway الخارجي وهو 98.10.10.5 عبر الـRouter ويقوم الـRouter بإضافة Record إلى NAT Table الخاصة به إذ يتم وضع الـOriginal Source Port And IP في ذلك الـRecord لتبين أن صاحب الـPacket المرسل هو الـClient الذي يملك العنوان المحدد ومن ثم يتم تغيير الـSource Port And IP الأصلي للـPacket إلى الـPublic IP الخاص بالـRouter و يتم توليد Port افتراضي بحيث يكون فريد على مستوى الشبكة ويوضع مكان الـSource Port في الـPacket ويضاف إلى الـ NAT Table Record ليتم استخراجه لاحقا عند مجيء الـResponse وبعد ذلك ترسل إلى الـGateway الخارجي.

7- يقوم الـGateway الخارجي بتوصيل الـHTTP Request إلى الـHTTP Server المعني ثم يقوم الـHTTP Server بإرسال الـResponse إلى الـRouter صاحب الـRequest ويحتوي ذلك الـResponse في الـDestination IP And Port على عنوان الـ Source IP And Port الموجود في الـRequest.

8- يستلم الـRouter ذلك الـResponse ويقوم باستخراج الـPrivate IP And Port من خلال الـDestination IP And Port الموجودة في الـResponse Packet والتي تم تخزينها في الـNAT Table عند عمل الـRequest وفي هذه المرحلة سيتمكن الـRouter من معرفة الجهاز صاحب الطلب الأصلي ويتم وضع الـPrivate Port And IP في الـDestination للـPacket ثم يتم توجيهها إلى الحاسب صاحب الطلب وهكذا تستطيع جميع الأجهزة الموجودة على الشبكة الداخلية المشاركة بـPublic IP وحيد واستخدامه لدخول إلى شبكة الإنترنت.

TCP / UDP Connection Establishment : 1.2

تمر عملية إنشاء الاتصال بمجموعة من المراحل، وفي العادة يقوم الطرف المرسل بإرسال طلب إنشاء الاتصال إلى الطرف الآخر وعند الموافقة على إجراء الاتصال يتم البدء بإرسال مجموعة من المعلومات إلى الطرف المستقبل، في بروتوكول الـ TCP تتم هذه العملية بثلاثة مراحل تسمى Three-way hand-shake وكما هو واضح في الشكل 1.12.



الشكل 1.12 ويبين الـThree Way hand-shake الخاصة ببروتوكول الـTCP

1- يقوم الطرف المرسل بتوليد رقم تسلسلي Sequence Number ويرسله إلى الـServer ويكون هذا الرقم المولد نقطة البدء لعملية الإرسال بحيث يتم زيادته بمقدار واحد بعد كل Acknowledgment.

2- يستلم الطرف المقابل الـSequence Number ويقوم بإرسال Acknowledgment إلى المرسل مضاف إليه الرقم التسلسلي الذي تم إرساله.

3- عند هذه المرحلة يكون قد تم الموافقة على بدأ الجلسة وعندها يقوم بإرسال طلبه مرفق معه الـAcknowledgment الذي أرسل من قبل المستقبل.

تمر هذه المراحل الثلاث عند الاتصال من خلال بروتوكول الـTCP أما في الـ UDP فيتم الإرسال مباشرة وبدون المرور بالعمليات الثلاث السابقة ويتم التراسل بدون إرسال Acknowledgments ولا يتحقق الـUDP من عمليات الوصول كما هو الحال في الـTCP وهو ما سيتم توضيحه في الجزء التالي من هذا الفصل.

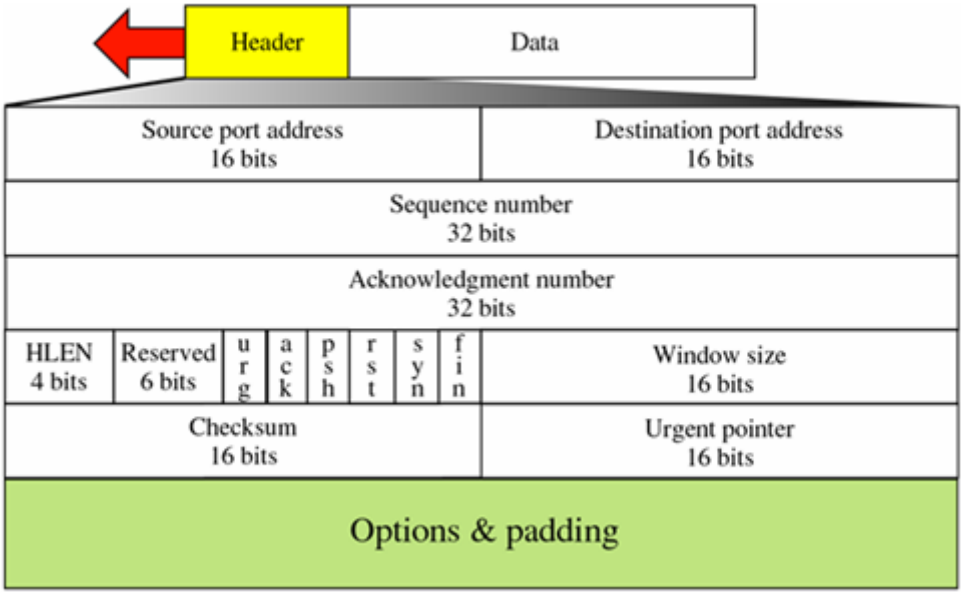
TCP & UDP Header Encapsulation : 1.3

في طبقة الـTransport Layer يتم التعامل مع إحدى البروتوكولين الـ TCP أو الـUDP حيث سنحدد فيها طبيعة الإرسال فإذا كان المطلوب هو الإرسال كـStream ونوع الإرسال هو Unicast فيتم اختيار الـTCP أما في حالة كان المطلوب هو الإرسال كـBroadcast أو Multicast فيتم اختيار الـUDP لعملية الإرسال لأن الـTCP غير قادر على الإرسال إلا لطرف وحيد يتصل معه بشكل Connection Oriented وكما لا يستطيع الـUDP إرسال البيانات بشكل Stream بدون وجود بروتوكول مساند كالـ RTP-Real Time Protocol والذي سنأتي على شرحه في الفصول اللاحقة ونستنتج من ذلك أن الـUDP يستطيع الإرسال

كـ Unicast و Multicast و Broadcast ولكن بشكل Datagram وليس بشكل Stream ويستطيع الـTCP الإرسال فقط كـUnicast Stream ويستطيع الـ SCTP Stream Control Transport Protocol (بروتوكول جديد غير مدعم من قبل Microsoft بشكل افتراضي على الـTCP/IP Stack الخاص بها يتميز بقدرته على التراسل كـMulticast بنفس ميزات بروتوكول الـTCP) الإرسال كـ Unicast و Multicast Stream ، وسوف نبين في الجزء التالي من هذا الفصل مبدأ عملية الإرسال باستخدام بروتوكول الـTCP وبروتوكول الـUDP وفي فصول لاحقة سيتم التطرق إلى البروتوكول المساند للـUDP وهو RTP.

TCP Encapsulation : 1.3.1

يتميز هذا البروتوكول بدعمه لكل عمليات التحكم سواء على مستوى الـ Data Flow أو حجم الـBuffer كما يدعم عمليات التحقق من الوصول وفق الترتيب السليم Delivered on Sequence وهذا واضح من تركيب الـ Header الخاصة به لاحظ الشكل 1.13:



الشكل 1.13 ويبين الـTCP Header

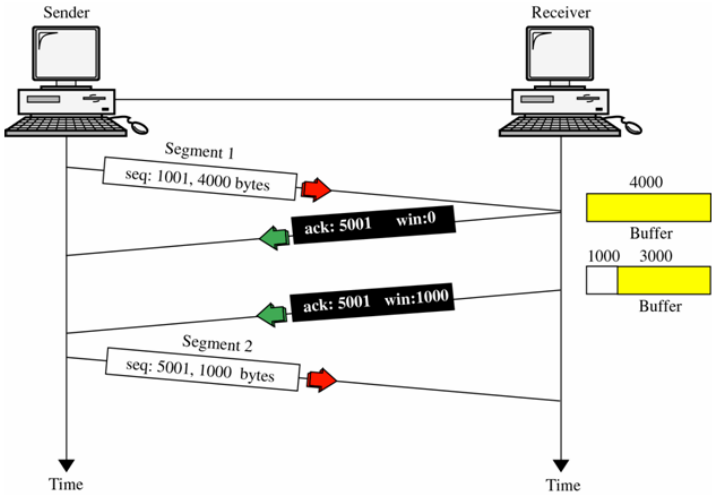
1.3.1.1 Source & Destination Port :

في المنطقة المخصصة بـ Port الإرسال والاستقبال ، يتم تحديد الـSource Port والـ Destination Port الخاصة بالبرنامج الذي يجري عملية الاتصال ومن المعروف أنه لا يمكن لأكثر من برنامج استخدام نفس رقم الـPort لكن يمكن للبرنامج الواحد استخدام أكثر من Port Address ، وبتأكيد فإن عملية اختيار الـPort ليست عشوائية إذ يجب الابتعاد عن الأرقام التي تبدأ بـ 1 وتنتهي 1023 إذ أنها أرقام لبروتوكولات معروفة في نظام التشغيل ومن الأمثلة عليها بروتوكول الـHTTP و الـ FTP وغيره وكما هو مبين في الجدول 1.1 السابق. ويبين الشكل 1.14 كيف أن الـPorts مقسمة إلى ثلاثة أقسام رئيسية القسم الأول وهو لبروتوكولات معروفة Well Known كما وضحنا سابقا والقسم الثاني مخصص لتطبيقات عالمية يتم تسجيلها في منظمة الـIANA تستخدم مجموعة من الـPorts كمثال الـ SQL Server والقسم الثالث مخصص لاختيار نظام التشغيل وهو Dynamic Range يختار منه نظام التشغيل Port محدد يستخدم كـ Source Port في عملية التراسل بحيث يكون فريد على مستوى الـApplications الأخرى.

الشكل 1.14 ويبين تقسيمات الـPorts حسب منظمة الـIANA الدولية

1.3.1.2 Sequence & Acknowledgment Number :

ويحتوي كل منهما على 32 Bits ويدل هذا الرقم على رقم التسلسلي للـSegment عند إرساله أو استقباله ويتم توليده عشوائيا عند بداية الاتصال أما رقم الـ Acknowledgment فيحتوي على الرقم التسلسلي للـ Segment و الذي تم التأكد من وصوله وتتم هذه العملية كما في الشكل 1.15:



الشكل 1.15 ويبين عملية التراسل من خلال الـTCP

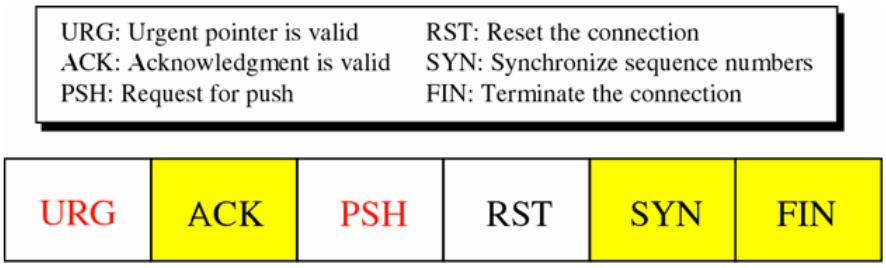
1.3.1.3 Header Length & Flags Controls :

ويحتوي الجزء الثاني من الـHeader الخاص بالـTCP على 32 Bits مقسمة إلى الـ Windows Size 16 Bits و 16Bits أخرى للـ Header Length + Flags Controls كما هو موضح في الشكل 1.16:

HLEN 4 bits	Reserved 6 bits	urg	ack	psh	rst	syn	fin	Window size 16 bits

الشكل 1.16 ويبين تقسيمات الـHeader Length في بروتوكول الـTCP

ويحتوي الـHeader Length على حجم الـHeader الخاص بالـTCP مقسوم على 4 أي لمعرفة حجم الـHeader نضرب الـHLEN (Header Length) بـ4 ، أما الـFlags Controls فهي 6 Controls تأخذ كل منها 1 Bit فإذا كانت قيمته 0 فهذا يعني أن هذه الخاصية غير مستخدمة وإذا كانت 1 فتعني أن الخاصية مستخدمة ويوضح الشكل 1.17 هذه الـFlags.



الشكل 1.17 ويبين الـFlags المستخدمة في بروتوكول الـTCP

1.3.1.4 Window Size :

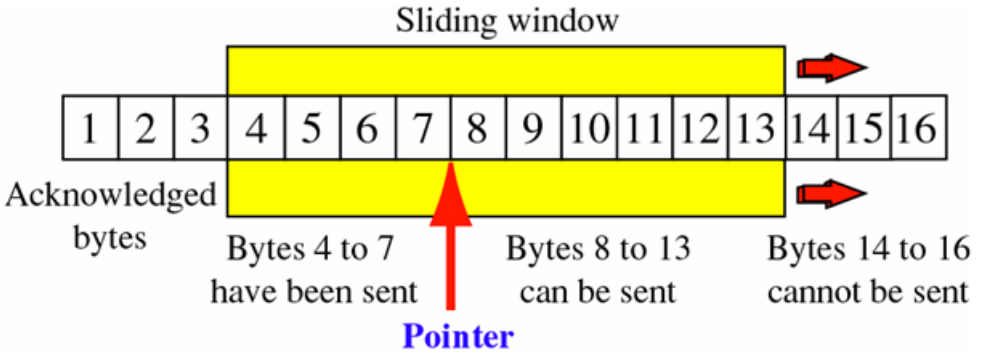
ويعرف فيه حجم الـSegment الذي يمكن إرساله من خلال الشبكة بناء على سرعة الوصول بين كل SYN Packet و ACK Packet، أي الوقت المستغرق لعملية التوصيل لكل Segment وقد تزيد أو تنقص بناءا على أدائية الشبكة.

1.3.1.5 Check Sum :

وهي 16 Bits وتستخدم لعملية التحقق من وصول الـTCP Header بشكل السليم حيث يتم جمع كافة قيم الـTCP Header (كل 16 Bits لوحدها) ثم قلبها ووضع الناتج في الـCheck Sum وفي الطرف المستقبل يقوم بتأكد من الـChecksum بضرب قيمة الـHLEN بـ4 ثم مقارنة الناتج مع مقلوب الـChecksum لمعرفة تفاصيل أكثر عن طرق حساب الـChecksum Algorithms يمكن الرجوع إلى الرابط http://rfc.net/rfc1146.html.

1.3.1.6 Urgent Pointer :

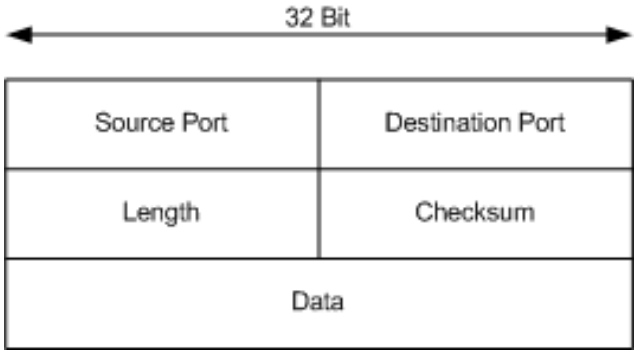
من المعروف أن الـData المرسلة عبر الـTCP يتم تجميعها في الـBuffer قبل أن يتم عرضها حيث يتم تحديد موقع الـData القادمة الجديدة في الـBuffer ومن هنا نحن بحاجة إلى وجود Pointer يؤشر على موقع الـData في الـBuffer وهو هنا الـUrgent Pointer لاحظ الشكل 1.18 والذي يوضح عملية وضع Data قادمة جديدة إلى الـBuffer الخاص بالجهاز المستقبل ويمكن الرجوع إلى الرابط http://www.faqs.org/rfcs/rfc2859.html لمزيد من المعلومات عن الـTime Sliding Window في بروتوكول الـTCP.



الشكل 1.18 ويبين مبدأ عمل الـSliding Window في بروتوكول الـTCP

UDP-User Datagram Protocol- Encapsulation : 1.3.2

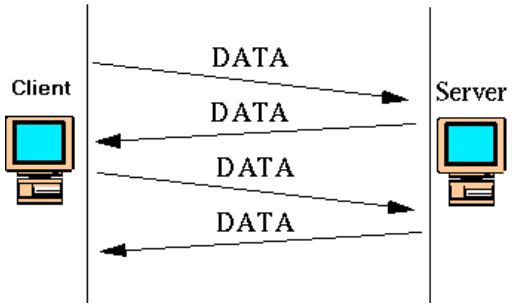
سوف نبين في هذا الجزء طبيعة الاتصال باستخدام الـUDP حيث يتميز هذا البروتوكول بإمكانية الإرسال كـ Multicast و Broadcast بالإضافة إلى الـUnicast بعكس الـTCP الذي يدعم الإرسال كـUnicast فقط وكما بينا سابقا، لكن مشكلة هذا البروتوكول هو عدم دعمه لعمليات التحكم على مستوى الـData Flow أو حجم الـBuffer كما لا يدعم عمليات التحقق من الوصول وفق الترتيب السليم Delivered on Sequence ويوضح الشكل 1.19 التركيب العام لهذا البروتوكول.



الشكل 1.19 ويبين مكونات الـHeader لبروتوكول الـUDP

يتم حساب الـCheck Sum و الـHeader Length بنفس طريقة الـTCP لكن لاحظ عدم وجود أي من الأمور الخاصة بالـBuffer Management أو الـDelivered On Sequence في الـHeader الخاص بالـUDP ، والمشكلة هنا أننا لا نستطيع عمل Fragmentation للـData حيث أن إعادة تجميعها

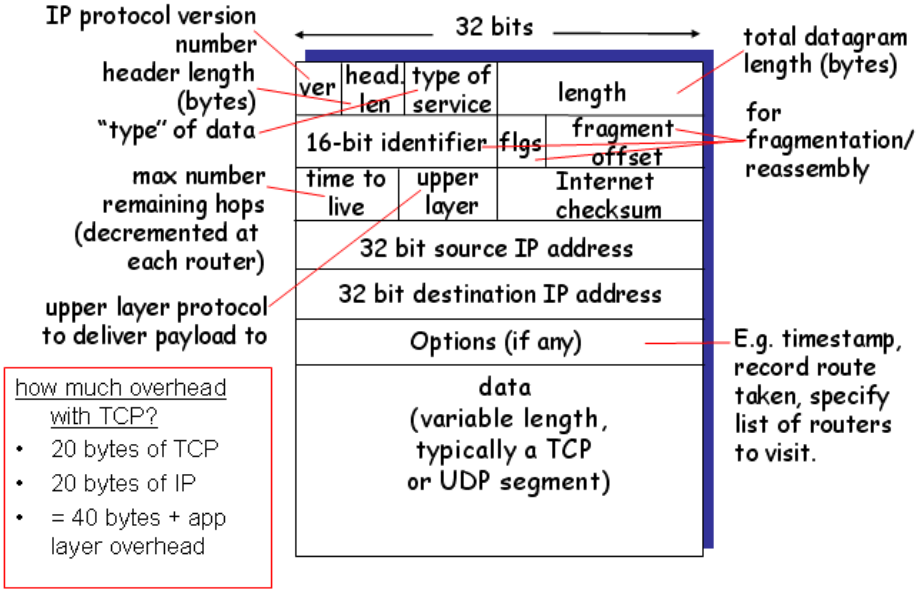
بالترتيب الصحيح أمر غير ممكن، كما أنه لا وجود لأي Acknowledgment لتحقق من وصول البيانات أو عدم وصولها ، لاحظ الشكل 1.20 والذي يوضح طريقة التراسل باستخدام الـUDP :



الشكل 1.20 ويبين طريقة التراسل بالـUDP حيث لا وجود للـAcknowledgment

1.4 : مقدمة في الـ IPv4 Architecture:

في الـNetwork Layer تتم عملية عنونة الـPacket باستخدام بروتوكول الإنترنت IP، وتتم هذه المرحلة بناءا على البروتوكول المستخدم في الـTransport Layer فإذا تم استخدام الـTCP عندها لا نستطيع إلا أن يكون العنوان المستخدم Unicast أما في حالة كان الـUDP هو البروتوكول المستخدم عندها نستطيع وضع عنوان Unicast أو Multicast أو Broadcast ويبين الشكل 1.21 التركيب العام لبروتوكول الإنترنت IPv4.



الشكل 1.21 ويبين التركيب العام لبروتوكول الـIPv4

1.4.1 مكونات الـIPv4 Header:

كما هو واضح في الشكل 1.21 السابق يتراوح حجم الـHeader الخاص بالـIPv4 من 20 إلى 60 Bytes بناءا على الـOptions المستخدم فإذا لم يتم استخدام الـOptions عندها سيكون حجم الـHeader ثابت وهو 20 Bytes وفي هذه الحالة يمكننا معرفة حجم الـData المرسلة عبر الـTCP كما يلي:

Data = total Length – (20 Bytes for IPv4 Header+ 20 Bytes For TCP Header)

وتقسم الـ20 Bytes الخاصة بالـIPv4 Header كما يلي:

4 Bits يوضع فيها الـVersion المستخدم وهو هنا IPv4 ويتم تمثيله 0100 في الـBinary.

4 Bits للـHeader Length ويوضع في حجم الـHeader مقسوم على 4

8 Bits للـType of Services حيث يمثل فيها مدى جودة الخدمة المطلوبة للبروتوكول المستخدم في الـApplication Layer ، ومنها Minimum Delay أو الـMaximum Services ولمزيد من المعلومات حول الـType Of Service في الـIP Header يمكن الرجوع إلى رابط منظمة الـIETF Internet Engineering Task Force التالي http://www.ietf.org/rfc/rfc0791.txt.

32 Bits وتستخدم لعمليات الـFragmentation وإعادة ترتيب الـFragments ولمزيد من المعلومات يمكن الرجوع إلى الرابط السابق للـIETF.

8 Bits وتستخدم لتحديد الـ TTL – Time to Live ويعبر عن عدد الـHops أو الـRouters التي يسمح للـPacket المرور من خلالها إلى أن تنتهي ، والعدد الافتراضي لها 16 hops وعند مرورها بكل Hop يتم طرح قيمة 1 منها إلى أن تصل إلى الحد الأدنى وفي حالة وصولها إلى الحد الأخير ولم تصل بعد إلى الـDestination يقوم الـRouter الأخير بعمل Drop لها وبالتالي يتم اعتبار الـResponse على أنه Time Out ولمزيد من المعلومات يمكن الرجوع إلى الرابط السابق للـIETF.

8 Bits أخرى لتحديد نوع البروتوكول المستخدم في الـUpper Layer سواء TCP أو UDP ولمزيد من المعلومات يمكن الرجوع إلى الرابط السابق للـIETF.

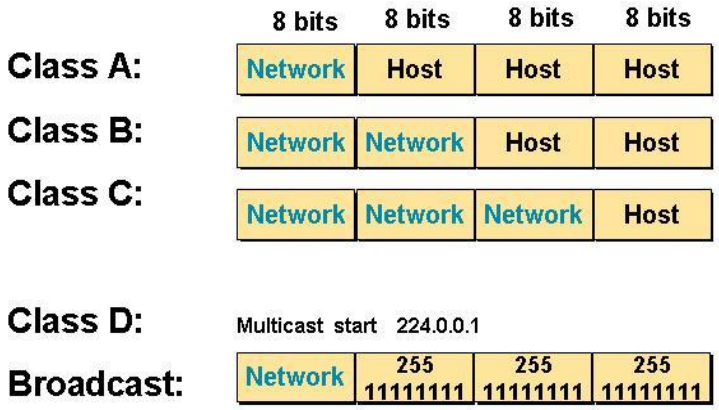
16 Bits للـ Checksum ولمزيد من المعلومات يمكن الرجوع إلى الرابط السابق للـIETF.

32 Bits لتحديد عنوان الجهاز المرسل Source IP Address.

32 Bits أخرى لتحديد عنوان الجهاز المستقبل Destination IP Address.

1.4.1.1 : طريقة تقسيم الـIP وفق Classful IP Address

تعتبر عملية العنونة باستخدام الـClassful بسيطة جدا ويقسم فيها العنوان إلى جزأين، يعبر الجزء الأول عن عنوان الـ Network ID والجزء الثاني عن الـHost ID وكما هو موضح في الشكل 1.22:

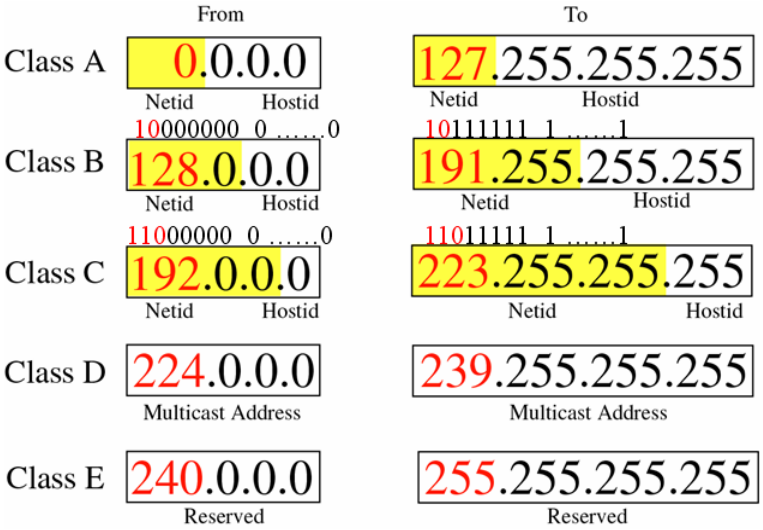


الشكل 1.22 ويبين طريقة الـClassful في الـIPv4

حيث يبدأ الـ Class A من 1 إلى 126 ويكون قيمة الـBit الأول في الـBinary صفر 0

ويبدأ Class B من 128 إلى 191 ويكون قيمة الـ Bit الأول في الـBinary واحد 1

ويبدأ Class C من 192 إلى 223 ويكون قيمة الـ Bit الأول والثاني في الـBinary واحد واحد 1 1 أما Class D فيبدأ من 224 وتكون الثلاثة Bits الأولى منه 111 ويستخدم لتعبير عن الـ Multicast IP ويبين الشكل 1.23 هذه التقسيمات.



الشكل 1.23 ويبين تقسيم الـUnicast والـBroadcast والـMulticast في الـIPv4

ولاستخراج عنوان الـ Broadcast من أي من التقسيمات السابقة يتم تعبئة الـBits الخاصة بالـHost ID بواحد وكمثال لمعرفة الـ Broadcast Address للعنوان 10.0.0.1 يجب أولا تحديد إلى أي Class ينتمي هذا العنوان، ومن الواضح أنه ينتمي إلى Class A لأن الجزء الخاص بالـ Network ID يبدأ بـ10 وهو بين 1 والـ126، إذا الجزء الخاص بالـHost ID هو 0.0.1 ولتحويله إلى Broadcast Address نضع قيمة 255 في الجزء الخاص بالـHost ID ويصبح العنوان كما يلي: 10.255.255.255 وهو الـ Broadcast Address للـ Class A في الـClassful Nation.

لكن المشكلة في الـClassful Nation أنه محدود إلى درجة كبيرة إذ أنه مناسب فقط لشبكات المحلية لكنه غير مناسب للإنترنت فمثلا القيمة العظمى لعدد العناوين للـ Class A هو 2^{32-8} Hosts ، وكان الحل بإمكانية دمج الـSubnets لجعل إمكانية توليد عناوين أكثر للـHost Part وهو ما يسمى بالـClassless.

1.4.1.2: طريقة التقسيم وفق CIDR - Classless InterDomain Routing IP Address:

وتتم هذه العملية باستخدام مجموعة من الـBits الخاصة بالـNetwork Part وضمها إلى الـ Host Part وكمثال لتوليد 1024 عنوان جديد من العنوان 192.168.1.0/24

نقوم باستئجار 2 Bits من الـ24 Bits الخاصة بالـNetwork ID عندها يصبح الـSubnet كما يلي : 192.168.1.0/22 ويصبح للـ Host ID ، 10 Bits = 22-32 أي 2^{10} Addresses وكما هو واضح في الشكل 1.24.

الشكل 1.24 ويبين كيفية استخدام Bits إضافية من الـNetwork Part وضمها لصالح Host Part لزيادة عدد العناوين في الـIPv4

ولمعرفة الـ Broadcast الخاص بالعنوان الجديد 192.168.1.0/22 نقوم بتقسيم الـ 22 Bits على العنوان حيث كل جزء يأخذ 8 Bits وكما يلي:

8	8	6	
192	168	1	0
11111111	11111111	11111100	00000000

ثم نحول الـ one's في الجزء الثالث إلى عشري وسيكون في المثال 252 وهذا يعني أن الـAddresses Range ستبدأ بـ 192.168.252.0 وستنهي بـ192.168.255.254

إذا سيكون الـ Broadcast IP هو : 192.168.255.255.

لاحظ أن مشكلة المحدودية للعناوين قد حلت بشكل جزئي باستخدام الـCIDR لكن مازالت الإمكانيات محدودة إذا أردنا توليد عناوين لملايين أو مليارات من الأجهزة والاستغناء عن تقنيات مشاركة الـPublic IP كما في حالة الـNAT وكان الحل في الـIPv6 حيث زاد فيه حجم العناوين من 32 Bits إلى 128 Bits.

1.4.2: IPv6 Architecture:

الـIPv6 وهو الجيل التالي للـIPv4 حل الـIPv6 الكثير من المشاكل التي تواجه الـIPv4 ونلخصها بمجموعة من النقاط:

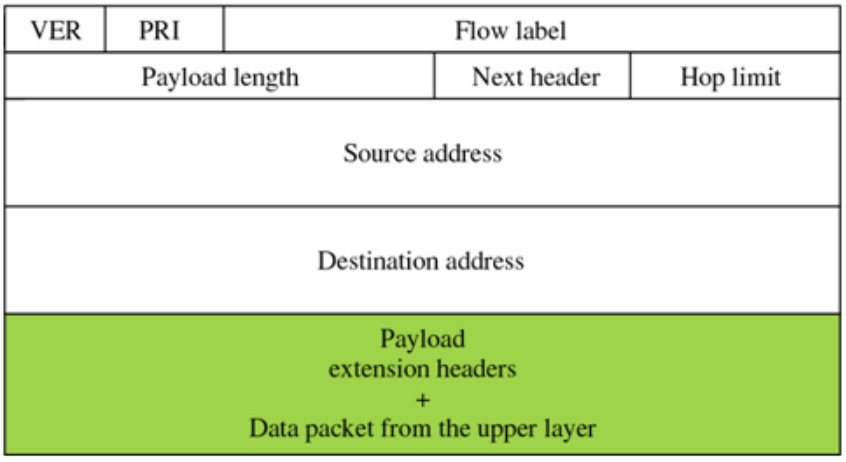
- الحاجة إلى عناوين أكثر حيث أن القيمة العظمى للـIPv4 Address تبقى محدودة مع زيادة الطلب على عناوين الـIP's حول العالم.

- قد يؤدي الـIPv4 إلى مجموعة من المشاكل وخاصة في Routing Table والتي قد تم حلها في الـIPv6

- مشكلة الـSecurity في الـIPv4 حيث لم يدعم أي من عمليات التشفير والتحقق Authentication على مستوى الـNetwork Layer وقد حلت هذه المشكلة في الـIPv4 باستخدام بروتوكول الـIPSec حيث يتم تشفير الـIP والـPort على مستوى الـSocket لكن أصبحت الحاجة ملحة لجعل هذه الـSecurity مدمجة على مستوى الـNetwork Layer .

- تطوير مبدأ الـBroadcast حيث تم تطويره إلى الـany cast إذ يتم إرسال رسالة واحدة إلى كل جهاز على الشبكة وفي حالة وجد الجهاز المعني يتوقف الـRouter عن الإرسال ، والهدف في هذه الطريقة إيجاد طريقة تخفض من الـBandwidth المستخدم عند البحث عن جهاز ما على الشبكة.

حل الـIPv6 كل هذه المشاكل حيث دعم ما يعادل 2^{128} Addresses وهو رقم كبير جدا كما قلل من حجم الـRouting Table مما سرع عمليات التوجيه Routing كما دعم عمليات الـAuthentication والتشفير على مستوى الـNetwork Layer كما تم حذف الـType of Services وحل محلها الـPriority أي الأولويات، ويوضح الشكل 1.25 مكونات الـHeader الخاصة بالـIPv6 :

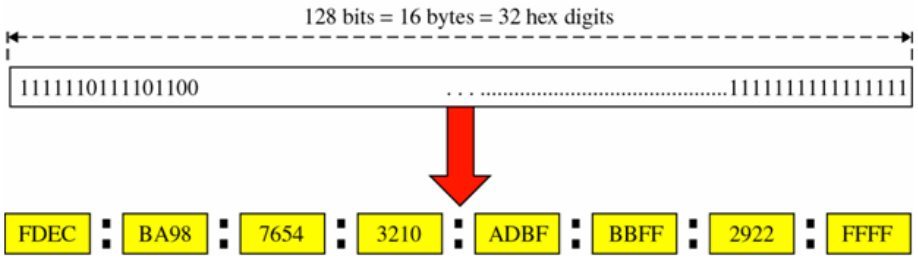


الشكل 1.25 ويبين مكونات الـHeader الخاص بالـIPv6

يستخدم الـIPv6 الـ hexadecimal بدلا من الـDecimal في الـIPv4 لتمثيل العنوان وتكون الصيغة العامة له كما يلي كمثال:

69dc:8864:ffff:ffff:0:1280:8c0a:ffff

لاحظ انه يتكون من ثمانية منازل بدلا من 4 بالـIPv4 ، في كل منزلة يوضع بها 16 Bits، ولتمثيله في الـHexadecimal نعطي 4 Digits لكل منزلة فيه وكما في الشكل 1.26.



الشكل 1.26 ويبين طريقة تمثيل عناوين الـIPv6 في الـHexadecimal

وتقسم العناوين في الـIPv6 إلى نوعين Provider-Based Addresses و Geographic-based Addresses حيث يقوم الـISP's بتوزيع العناوين على الـClients باستخدام الـStandard الخاص Provider-Based Addresses وأما الـ Geographic-based Addresses فهو مخصص لإعطاء العناوين الدولية ، أي انه سيكون لكل دولة رمز خاص يكون في بداية العنوان وكما يلي:

- Provider-Based Addresses

a) Registry ID

b) Provider ID

c) Subscriber ID

d) Subscriber Subnet

e) Host Number

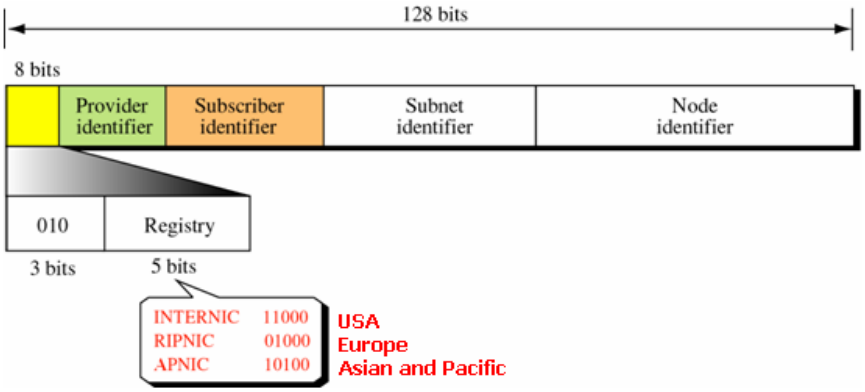
- Geographic-based Addresses

a) Registry ID

b) World Zone

c) Country, City, etc.

لاحظ الشكل التالي:



الشكل 1.27 ويبين طريقة تقسيم الـIPv6 Address بناء على الدولة في حالة العناوين المخصصة للـISP's

لكن لم يتم إلى الآن اعتماد نظام الـIPv6 في الـISP's حول العالم إذ سيبقى على ما هو عليه إلى حين تغيير البنية التحتية للمزودين والزبائن وهو ما سيحتاج إلى ميزانيات ضخمة وكذلك وقت كبير لتبنيه بشكل كامل. لمزيد من المعلومات حول الـIPv6 يمكن الرجوع إلى الرابط http://www.ietf.org/rfc/rfc2460.txt لمنظمة Internet Engineering Task Force IETF.

الخلاصة:

بينا في هذا الفصل مقدمة عن بروتوكول الـTCP/IP بشرح المراحل التي تمر بها عملية الاتصال في كل طبقة من طبقات الـTCP/IP Model وكذلك شرح لمكونات وكيفية عمل البروتوكولات في كل طبقة من تلك الطبقات وأهمها كالـHTTP والـDNS في الـApplication Layer والـTCP والـUDP في الـTransport Layer والـIPv4 و الـIPv6 في الـNetwork Layer والـARP والـRARP في الـData Link Layer.