

Contenido

| | |
|------------------------------|----------|
| Arquitectura de Redes | 1 |
| Teoría | 2 |
| General | 2 |
| Capas | 2 |
| Tamaños | 2 |
| Diagramas de flujo | 3 |
| Internet | 4 |
| Serial | 4 |
| Resumen headers | 4 |
| Ruteo | 5 |
| Ruteo dinamico | 5 |
| Route redistribution | 6 |
| Ethernet | 6 |
| Header | 6 |
| MTU | 6 |
| Cableado | 6 |
| Normas | 6 |
| Switch | 7 |
| ARP | 7 |
| Header | 7 |
| Funcionamiento | 8 |
| Capturas | 8 |
| ARP gratuitous | 9 |
| IPv4 | 10 |
| Header | 10 |
| Asignación | 11 |
| Tipos de direcciones | 11 |
| Clases | 11 |
| Redes privadas | 12 |
| Redes reservadas | 12 |
| VLSM y CIDR | 12 |
| Máscaras comunes | 14 |
| Fragmentación | 14 |
| Traceroute | 14 |
| ICMPv4 | 16 |
| Header | 16 |
| Ping | 17 |

| | |
|----------------------|----|
| Otros usos | 17 |
| Redirección ICMP | 17 |
| Capturas | 17 |
| NAT | 19 |
| Basic NAT | 19 |
| NAPT | 19 |
| Tipos | 19 |
| Otras terminologías | 19 |
| DHCP | 19 |
| Header | 19 |
| Opciones | 20 |
| Funcionamiento | 20 |
| Estados | 22 |
| DHCP relaying | 23 |
| Capturas | 23 |
| HDLC | 28 |
| Control de flujo | 28 |
| Métodos | 28 |
| Detección de errores | 29 |
| Tramas | 29 |
| Header | 29 |
| Control | 29 |
| Fases de operación | 30 |
| Bit stuffing | 30 |
| PPP | 30 |
| Header | 31 |
| Estados | 31 |
| Control de enlace | 32 |
| Control de red | 32 |
| PPPoE | 32 |
| Estados | 32 |
| Capturas | 33 |
| Header | 34 |
| VLAN | 34 |
| 802.1Q | 34 |
| ISL | 35 |
| IPv6 | 35 |
| Header | 35 |
| Tipos de direcciones | 35 |
| SLAAC | 36 |
| Migración desde IPv4 | 36 |
| Fragmentación | 36 |

| | |
|-------------------------------------|----|
| ICMPv6 | 38 |
| Header | 38 |
| Ping | 38 |
| Header | 38 |
| NDP | 38 |
| Otros usos | 38 |
| Capturas | 39 |
| 6to4 | 41 |
| Direcciones | 42 |
| RIP | 42 |
| Bellman-Ford | 42 |
| RIPng | 43 |
| Captura | 43 |
| OSPF | 45 |
| Dijkstra | 45 |
| OSPFv3 | 45 |
| Captura | 45 |
| IGRP | 47 |
| BGP | 47 |
| Formas de aceptar rutas | 48 |
| Sincronización | 48 |
| Mi escenario en GNS3 | 48 |
| Rutas | 49 |
| Configuraciones | 50 |
| MPLS | 53 |
| Definiciones | 53 |
| Planos | 54 |
| Funcionamiento | 54 |
| Problemas a solucionar en el futuro | 54 |
| TCP | 55 |
| UDP | 55 |
| GNS3 | 55 |
| Elementos usados | 55 |
| Otros tips | 56 |
| Wireshark | 56 |
| Ejemplos de filtros | 56 |
| Comandos | 57 |
| Linux | 57 |
| Básico | 57 |
| ARP | 57 |
| NAT | 57 |
| Servidor DHCP | 57 |

| | |
|----------------|----|
| Cliente DHCP | 58 |
| Servidor PPPoE | 58 |
| Cliente PPPoE | 58 |
| IPv6 | 58 |
| Cisco | 59 |
| Básico | 59 |
| NAT | 60 |
| Servidor DHCP | 60 |
| Enlace serial | 61 |
| Servidor PPPoE | 61 |
| VLAN (switch) | 62 |
| VLAN (router) | 62 |
| IPv6 | 62 |
| 6to4 | 63 |
| RIP | 63 |
| RIPv2 | 63 |
| RIPng | 64 |
| OSPF | 64 |
| OSPFv2 | 64 |
| OSPFv3 | 65 |
| BGP | 65 |
| MPLS | 66 |
| Mikrotik | 67 |
| NAT | 68 |
| Servidor DHCP | 68 |
| Servidor PPPoE | 68 |
| IPv6 | 68 |
| 6to4 | 68 |
| RIP | 69 |
| RIPv2 | 69 |
| RIPng | 69 |
| OSPF | 69 |
| OSPFv2 | 69 |
| OSPFv3 | 70 |
| Links | 71 |

Arquitectura de Redes

Teoría

General

Por hacer

Reordenar y poner cosas en otros documentos. Ver como separar IPV4 de IPV6.

Capas

1. Capa física.

- Unidad: Símbolos, Bits.

2. Capa de enlace.

- Unidad: Trama.
- Protocolos: Ethernet, PPP, HDLC, PPPoE (aunque esté dentro de Ethernet).

3. Capa de red.

- Unidad: Paquete.
- Protocolos: IP, ARP, ICMP (aunque esté dentro de IP).

4. Capa de transporte.

- Unidad: Segmento (TCP), Datagrama (UDP).
- Protocolos: TCP, UDP

5. Capa de aplicación.

- Unidad: Datos
- Protocolos: DHCP, HTTP.

| OSI | TCP/IP | Lo que usamos |
|-----------------|-----------------|---------------|
| 7. Aplicación | Aplicación | 5. Aplicación |
| 6. Presentación | | |
| 5. Sesión | | |
| 4. Transporte | Transporte | 4. Transporte |
| 3. Red | Internet | 3. Red |
| 2. Enlace | Acceso a la red | 2. Enlace |
| 1. Física | | 1. Física |

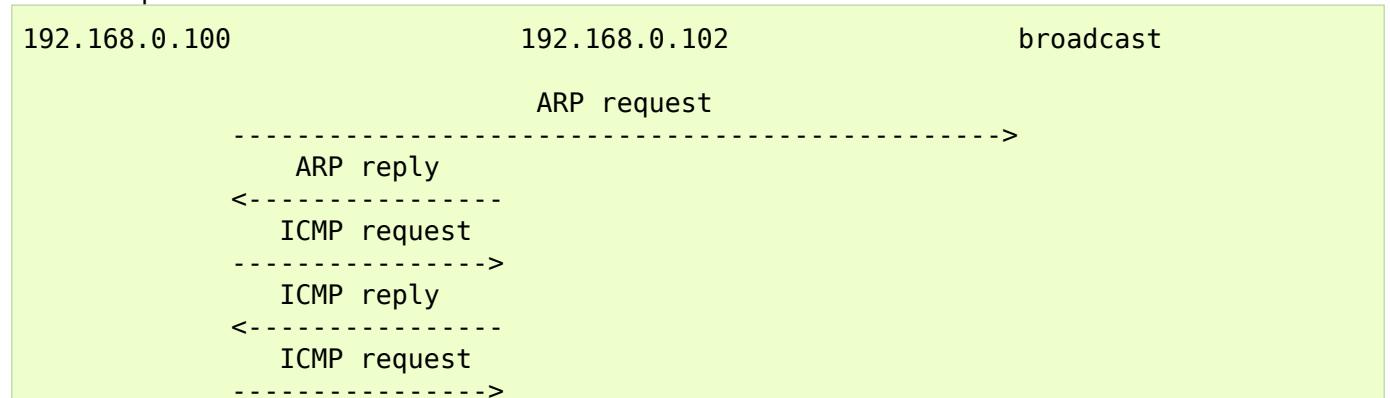
Tamaños

- Tamaños de cabeceras
- Ethernet:

- Sin contar preámbulo ni CRC: 14 bytes.
 - Sin contar preámbulo y contando CRC: 18 bytes.
 - Contando preámbulo y CRC: 26 bytes.
 - Si incluye tags (VLAN): + 4 bytes.
 - ARP: 28 bytes.
 - IP: 20 bytes.
 - ICMP: 8 bytes.
 - PPPoE: 6 bytes.
 - PPP:
 - Sin contar flags (delimitadores) ni FCS: 4 bytes.
 - Sin contar flags (delimitadores) pero sí FCS: 6 bytes.
 - Contando flags (delimitadores) y FCS: 8 bytes.
 - Dentro de PPPoE solo lleva el campo Protocol: 2 bytes.
 - DHCP:
 - Sin contar SNAME, ni FILE, ni opciones: 44 bytes.
- Tamaños de tramas:
- Longitud total máxima de datos que puede llevar Ethernet es 1500 bytes, que sumado al header, CRC y preambulo Ethernet dan 1526 bytes por trama.
 - Restando los 20 bytes de header IP, la cantidad máxima que puede llevar IP es 1480 bytes. Esto es suponiendo que no hay opciones en el header, en tal caso éstas opciones pueden ocupar hasta 32 bytes más.
 - Las tramas Ethernet que llevan un poco más de 1500 bytes se llaman **baby giant**.
 - Los **jumbo frames** Ethernet llevan hasta 9000 bytes de datos, son soportados por placas Gigabit, pero la mayoría de los ISP no los aceptan.
 - PPPoE:
 - PPPoE va dentro de Ethernet, después del header PPPoE va el header PPP y después va el paquete IP. Por lo tanto, si el MTU de Ethernet es 1500, lo máximo que puede llevarse de datos es $1500 - 6 - 2 = 1492$ bytes.
 - Dentro de PPPoE va IP, por lo tanto hay que restar 20 bytes más para saber cuántos bytes entran en IP: 1472 bytes.
 - Al usar VLAN, el header Ethernet aumenta en 4 bytes pero sigue teniendo un MTU de 1500.

Diagramas de flujo

ICMP con primero un ARP:



ICMP reply

<-----

Internet

Algunas definiciones:

- ISP: Internet Service Provider.
- NSP: National Service Provider.
- NAP: Network Access Point.
- CABASE: Cámara Argentina de Internet.
- ICANN: Internet Corporation for Assigned Names and Numbers, coordina la asignación de DNS, direcciones IP y sistemas autónomos.
- IANA: Internet Assigned Numbers Authority, es la autoridad que asigna nombres de dominio, direcciones IP y sistemas autónomos. En este momento a esta función la posee el ICANN.
- RIR: Regional Internet Registry, organización que asigna direcciones IP y números de sistemas autónomos en una región:
 - ARIN: Estados Unidos, Canadá.
 - RIPE NCC: Europa, medio oriente y Asia central
 - APNIC: Asia y Pacífico.
 - LACNIC: Latin American and Caribbean Internet Addresses Registry.
 - AfriNIC: África.
- AS: Sistema Autónomo, un conjunto de routers y redes bajo una misma administración, se identifican por un número de 16 bits.
- IETF: Internet Engineering Task Force, se encarga de crear los estándares RFC.
- RFC: Request For Comments, para que un protocolo se estandarice tiene que estar publicado en una RFC, pero no todos los RFC son estándares. Son publicados por la IETF.

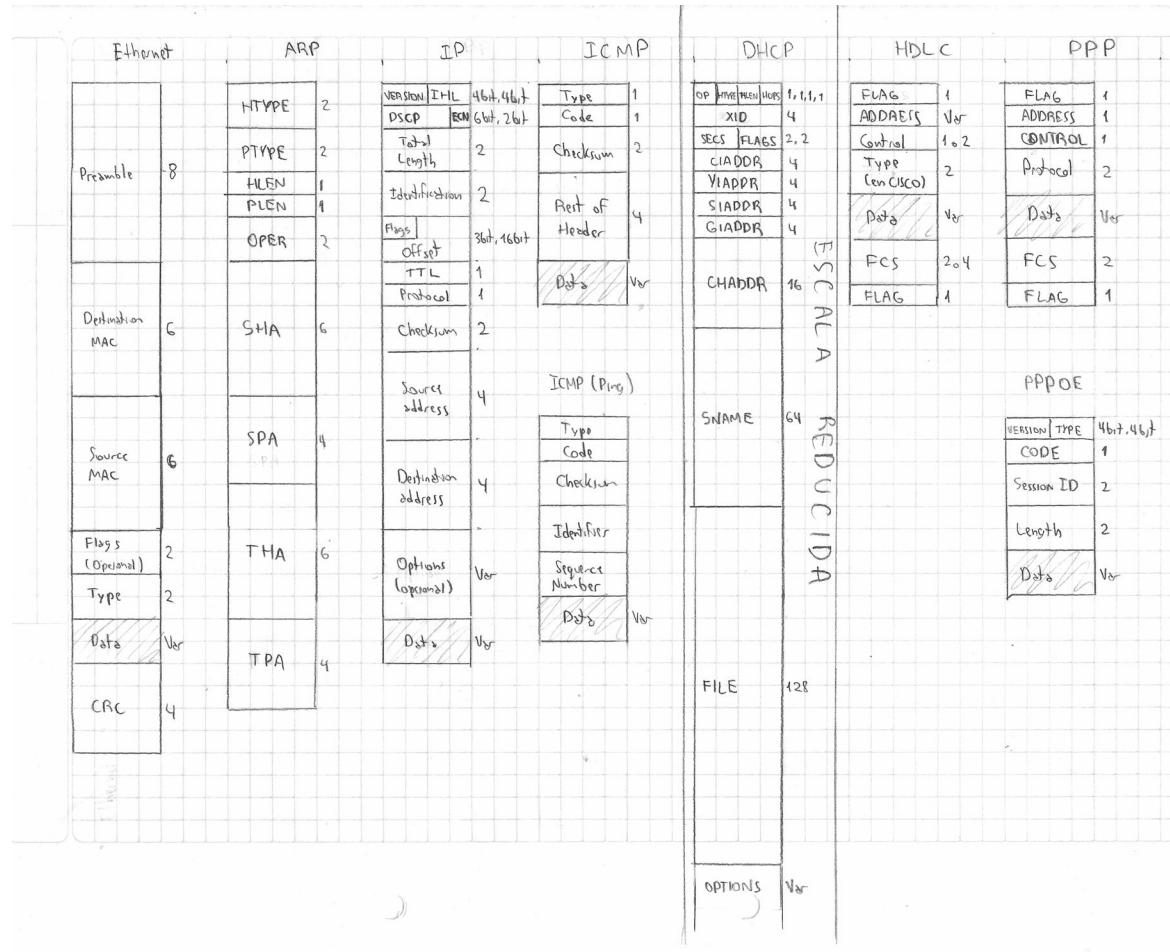
Serial

Se puede usar HDLC o PPPoE. El cable tiene dos extremos:

- DSE: (Hembra), Configura velocidad.
- DTE: (Macho).

Resumen headers

Para comparar un poco los tamaños, todos ocupan 1 byte por renglón, excepto DHCP que es tan grande que lo dibujé a 4 bytes por renglón.



Ruteo

- Una ruta es un prefijo más el próximo salto.
- Hay dos funciones básicas:
 - RIB (Routing Information Base): Para determinación de ruta.
 - FIB (Forwarding Information Base): Para conmutación de paquetes, la información que tiene deriva de las mejores rutas de la RIB.
- Las tablas de ruteo indican el próximo salto. Si hay varias rutas que contienen a la dirección de destino, se elige la ruta que tiene la máscara más larga. Si tienen la misma longitud de máscara se elige la ruta con menor distancia administrativa que sólo tiene significado local.
- Los algoritmos clasifican las rutas de acuerdo a métricas, que son números generados a partir de variables como cantidad de saltos, velocidad de enlaces, etc.
- Pueden ser protocolos

Ruteo dinámico

- Pueden ser IGP (Interior Gateway Protocol) o EGP (External Gateway Protocol) dependiendo de si es interno a un sistema autónomo o no. El único EGP es BGP.
- La distancia administrativa (grado de conocimiento y confiabilidad) depende del protocolo, tiene un valor por defecto pero se puede cambiar:
 - RIP: 120.
 - Redes directamente conectadas: 0.
 - Rutas estáticas: 1.

- eBGP: 20.
- OSPF: 110.
- IS-IS: 115.
- iBGP: 200.
- Un protocolo elige entre distintos caminos para llegar a un destino mirando las métricas.
- Si hay una ruta aprendida por más de un protocolo, se elige la ruta con menor distancia administrativa.
- Por como funcionan los sistemas operativos de routers, en protocolos como RIP o OSPF hay dos formas de propagar rutas directamente conectadas: usando comandos como network en todas las redes y pasivando algunas si es necesario, o usando network en las activas y anunciando las demás con un comando como redistribute connected.

Route redistribution

- Para propagar rutas de un protocolo de ruteo dinámico a otro protocolo se usa *Route Redistribution*, no es un estándar y varía un poco dependiendo de los fabricantes.
- En el caso de que hayan dos routers de borde, pueden ocurrir loops. Por lo tanto una ruta recibida desde un protocolo no debe ser reinyectada en el mismo protocolo.
- Creo que este tema es complejo cuando se redistribuye entre dos IGP. En cambio cuando se redistribuye entre BGP y un IGP debería ser más estándar y fácil.

Ethernet

- Está el Ethernet II que es lo que se usa, sino también está el Ethernet 802.3 que tiene un header ligeramente distinto.

Header

- Preamble (8 bytes): Generalmente no se ve en Wireshark.
- Destination MAC address (6 bytes).
- Source MAC address (6 bytes).
- Tags (Opcional) (4 bytes): En el caso en el que se utilicen VLANs.
- Type (2 bytes): ID del protocolo encapsulado.
- Data (46-1500 bytes).
- Frame Check Sequence (FCS) (4 bytes): Es CRC, generalmente no se ve en Wireshark.

MTU

Ver en la sección General.

Cableado

Hay dos normas para el orden de los cables, 568-A y 568-B. Se puede usar cualquiera de las dos normas.

- Si las normas de ambos extremos son iguales, es un cable directo. Sirve para conectar dispositivos de distintas capas (PC-Switch, Switch-Router, etc.).
- Si las normas de ambos extremos son distintas, es un cable cruzado. Sirve para conectar dispositivos de la misma capa (PC-PC, Router-Router, etc.).

Normas

Este es el orden de los colores visto desde la cara opuesta a la pestaña, con el cable apuntando hacia arriba y viendo los colores de izquierda a derecha.

- Norma 586-A:

- Verde y blanco. TX+.
- Verde. TX-.
- Naranja y blanco. RX+.
- Azul.
- Azul y blanco.
- Naranja. RX-.
- Marrón y blanco.
- Marrón.

- Norma 586-B:

- Naranja y blanco. TX+.
- Naranja. TX-.
- Verde y blanco. RX+.
- Azul.
- Azul y blanco.
- Verde. RX-.
- Marrón y blanco.
- Marrón.

Switch

El switch sólo reenvía las tramas unicast por una interfaz salvo de que se trate de una trama broadcast, en tal caso hace un *flood* reenviando por todas las bocas.

Para esto el switch tiene una tabla de direcciones MAC con su correspondiente interfaz. Cuando recibe una trama en alguna interfaz aprende la MAC mirando la dirección de origen. Un caso especial es cuando recibe una trama que tiene como destino una MAC que no está en la tabla, como no sabe por qué interfaz reenviar, entonces hace un *flood*, aprenderá esa MAC cuando ese host responda.

ARP

Address Resolution Protocol.

- Va sobre Ethernet.
- No debería ser de capa 3 porque no tiene ruteo, y no debería ser de capa 2 porque no es Ethernet. Así que no está bien definido.
- Se usa para descubrir la MAC de algún host a partir de su IP.
- Los demás hosts aprovechan a escuchar ARP de otros para aprender sus MAC, aunque supongo que lo hacen solamente si el destino es Broadcast en Ethernet.

Header

- Hardware type (HTYPE) (2 bytes): Protocolo de enlace, siempre es Ethernet (0x0001).
- Protocol type (PTYPE) (2 bytes): Protocolo de red, siempre es IP (0x0800).
- Hardware address length (HLEN) (1 byte): Largo de dirección MAC, siempre es 6.
- Protocol address length (PLEN) (1 byte): Largo de dirección IP, siempre es 4.

- Operation (OPER) (2 bytes):
 - 1: Request
 - 2: Reply
 - 3: Request RARP
 - 4: Reply RARP
- Sender hardware address (SHA) (6 bytes): MAC del emisor.
- Protocol hardware address (SPA) (4 bytes): IP del emisor.
- Target hardware address (THA) (6 bytes): MAC del destino.
- Target hardware address (TPA) (4 bytes): IP del destino.

Funcionamiento

Por ejemplo si A quiere saber la MAC de B:

- Request:
 - Ethernet:
 - Source MAC: A.
 - Destination MAC: FF:FF:FF:FF:FF:FF.
 - Type: 0x0806 (ARP).
 - ARP:
 - Opcode: 1 (Request).
 - Sender MAC: A.
 - Target MAC: 00:00:00:00:00:00.
 - Sender IP: A.
 - Target IP: B.
- Reply:
 - Ethernet:
 - Source MAC: B.
 - Destination MAC: A.
 - Type: 0x0806 (ARP).
 - ARP:
 - Opcode: 2 (Reply).
 - Sender MAC: B.
 - Target MAC: A.
 - Sender IP: B.
 - Target IP: A.

Capturas

Request:

```
Frame 75: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
Ethernet II, Src: LiteonTe_13:12:f4 (24:fd:52:13:12:f4),
    Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Destination: Broadcast (ff:ff:ff:ff:ff:ff)
Source: LiteonTe_13:12:f4 (24:fd:52:13:12:f4)
Type: ARP (0x0806)
```

```
Address Resolution Protocol (request)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
Sender MAC address: LiteonTe_13:12:f4 (24:fd:52:13:12:f4)
Sender IP address: 192.168.0.102
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.0.200
```

Reply:

```
Frame 1480: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
Ethernet II, Src: HonHaiPr_13:7f:55 (08:3e:8e:13:7f:55),
    Dst: Micro-St_9e:e5:e1 (d8:cb:8a:9e:e5:e1)
Destination: Micro-St_9e:e5:e1 (d8:cb:8a:9e:e5:e1)
Source: HonHaiPr_13:7f:55 (08:3e:8e:13:7f:55)
Type: ARP (0x0806)
Address Resolution Protocol (reply)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: HonHaiPr_13:7f:55 (08:3e:8e:13:7f:55)
Sender IP address: 192.168.0.20
Target MAC address: Micro-St_9e:e5:e1 (d8:cb:8a:9e:e5:e1)
Target IP address: 192.168.0.120
```

ARP gratuitous

A gratuitous ARP request is an AddressResolutionProtocol request packet where the source and destination IP are both set to the IP of the machine issuing the packet and the destination MAC is the broadcast address ff:ff:ff:ff:ff:ff. Ordinarily, no reply packet will occur. A gratuitous ARP reply is a reply to which no request has been made.

Para hacer ARP gratuitos:

```
arping -A -c 100000 -I enp4s0 {mi IP}
```

Captura al usar ese comando:

```
Frame 32: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on
    interface 0
Ethernet II, Src: Dell_17:2b:b2 (10:7d:1a:17:2b:b2), Dst: Broadcast
    (ff:ff:ff:ff:ff:ff)
Destination: Broadcast (ff:ff:ff:ff:ff:ff)
Source: Dell_17:2b:b2 (10:7d:1a:17:2b:b2)
Type: ARP (0x0806)
Padding: 0000000000000000000000000000000000000000000000000000000000000000
Address Resolution Protocol (reply/gratuitous ARP)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
[Is gratuitous: True]
Sender MAC address: Dell_17:2b:b2 (10:7d:1a:17:2b:b2)
Sender IP address: 192.168.1.215
Target MAC address: Dell_17:2b:b2 (10:7d:1a:17:2b:b2)
Target IP address: 192.168.1.215
```

En la casa del nano capturé ARP gratuitous distintos, algunos reply y otros request:

```
Frame 4: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on
         interface 0
Ethernet II, Src: Zhejiang_ad:9f:5f (4c:11:bf:ad:9f:5f), Dst: Broadcast
             (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request/gratuitous ARP)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    [Is gratuitous: True]
    Sender MAC address: Zhejiang_ad:9f:5f (4c:11:bf:ad:9f:5f)
    Sender IP address: 192.168.15.50
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.15.50

Frame 11: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on
         interface 0
Ethernet II, Src: Zhejiang_ad:9f:5f (4c:11:bf:ad:9f:5f), Dst: Broadcast
             (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (reply/gratuitous ARP)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    [Is gratuitous: True]
    Sender MAC address: Zhejiang_ad:9f:5f (4c:11:bf:ad:9f:5f)
    Sender IP address: 192.168.15.50
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.15.50
```

Hay algunos ARP gratuitous en donde es una request con la target MAC vacía con ceros, pidiendole al que tenga la misma IP que yo que me responda con su MAC.

IPv4

Internet Protocol.

- Va sobre Ethernet o PPPoE.
- Tiene direcciones de 32 bits.

Header

- Version (4 bits): Siempre es 4.
- Internet Header Length (IHL) (4 bits): Cantidad de palabras de 32 bits en el header, si no hay opciones es 5, que equivale a 20 bytes.
- Differentiated Services Code Point (DSCP) (6 bits): Tipo de servicio, usado para QoS.
- Explicit Congestion Notification (ECN) (2 bits): Se usa poco, indica que hay congestión en la red.
- Total Length (16 bits): Largo del header y datos en bytes.
- Identification (16 bits): Identifican un grupo de paquetes al fragmentar.
- Flags (3 bits): El bit 2 indica que vienen más fragmentos, el bit 1 indica que no se debe fragmentar este paquete, el bit 0 no se usa.
- Fragment offset (13 bits): Especifica el offset de este fragmento en particular, cuenta de a 8 bytes.

- Time to live (TTL) (8 bits): Cantidad de saltos hasta que el paquete se descarte, disminuye en cada salto.
- Protocol (8 bits): Identifica al protocolo encapsulado.
- Header checksum (16 bits): Checksum del header, cada vez que un router disminuye el TTL debe recalcular este checksum, esto es ineficiente.
- Source address (32 bits): IP de origen.
- Destination address (32 bits): IP de destino.
- Options (variable): Opciones que no se usan normalmente, se debe hacer un padding para que sea múltiplo de 32 bits.

Asignación

- La mitad superior del espacio de direcciones de clase A se reserva indefinidamente para tener la posibilidad de usarlo en la transición a un nuevo sistema de numeración.
- Las redes de clase B sólo se asignan a organizaciones que puedan probar claramente que las necesitan.
- Los requerimientos para las redes de clase B son que la organización solicitante:
 - Tenga un esquema de subnetting con más de 32 subredes dentro de su red operativa.
 - Tenga más de 4096 hosts.
- A las organizaciones que no satisfacen los requerimientos para una red de clase B se les asigna un bloque de redes clase C numeradas consecutivamente, para que se pueda hacer supernetting y rutear a todas juntas.
- La mitad inferior del espacio de direcciones de clase C (números de red del 192.0.0 al 223.255.245) se divide en 8 bloques que separa las autoridades regionales. Están reservadas del siguiente modo:
 - 192.0.0 - 193.255.255: Multiregional.
 - 194.0.0 - 195.255.255: Europa.
 - 196.0.0 - 197.255.255: Otros.
 - 198.0.0 - 199.255.255: Norte América.
 - 200.0.0 - 201.255.255: Centro y Sudamérica.
 - 202.0.0 - 203.255.255: Borde del Pacífico.
 - 204.0.0 - 205.255.255: Otros.
 - 206.0.0 - 207.255.255: Otros.
- La mitad superior del espacio de direcciones de clase C(208.0.0 a 223.255.255) permanece sin asignar y sin reservar.

Tipos de direcciones

Clases

- A:
 - Comienzan con: 0b0
 - Redes: 0.0.0.0 - 127.0.0.0
 - Redes privadas: 10.0.0.0
 - Numero de hosts: $2^{24} - 2$
- B:

- Comienzan con: 0b10
- Redes: 128.0.0.0 - 191.255.0.0
- Redes privadas: 172.16.0.0 - 172.31.0.0
- Numero de hosts: $2^{16} - 2$
- C:
 - Comienzan con: 0b110
 - Redes: 192.0.0.0 - 255.255.255.0
 - Redes privadas: 192.168.0.0 - 192.168.255.0
 - Numero de hosts: $2^8 - 2$
- D (multicast):
 - Redes: 224.0.0.0 - 239.255.255.0
- E (reservado):
 - Redes: 240.0.0.0 - 255.255.255.255

Redes privadas

- Clase A: 10.0.0.0
- Clase B: 172.16.0.0 - 172.31.0.0
- Clase C: 192.168.0.0 - 192.168.255.0

Redes reservadas

Estas son algunas pero hay más.

- 0.0.0.0 - 0.255.255.255: Para representar la red actual.
- 127.0.0.0 - 127.255.255.255: Para loopback.
- 169.254.0.0 - 169.254.255.255: para link-local.
- 192.0.2.0 - 192.0.2.255: TEST-NET-1, Para documentación y ejemplos.
- 198.51.100.0 - 198.51.100.255: TEST-NET-2, Para documentación y ejemplos.
- 203.0.113.0 - 203.0.113.255: TEST-NET-3, Para documentación y ejemplos.

VLSM y CIDR

Son VLSM (Variable Length Subnet Mask) y CIDR (Classless Inter-Domain Routing) reemplazan a la división de redes por clases y permiten crear redes con cualquier máscara de red. Los dos son mas o menos lo mismo pero técnicamente:

- CIDR: Permite que las direcciones dadas por la IANA no estén limitadas a ser clase A, B o C sino que tengan cualquier máscara.
- VLSM: Es una estrategia para subdividir una red en varias más chicas usando CIDR.

Los profes dicen que si uno agarra por ejemplo una /16, si se corre la máscara a la izquierda es CIDR/Subnetting, si se corre a la derecha es VLSM/Subnetting.

Carta de ayuda para hacer VLSM:

| /25 - .128 | | /26 - .192 | | /27 - .224 | | /28 - .240 | | /29 - .248 | | /30 - .252 | |
|-------------|-----------|------------|-----------|------------|-----------|------------|-----------|------------|-----------|------------|-----------|
| 128-2 hosts | | 64-2 hosts | | 32-2 hosts | | 16-2 hosts | | 8-2 hosts | | 4-2 hosts | |
| Net | Hosts | Net | Hosts | Net | Hosts | Net | Hosts | Net | Hosts | Net | Hosts |
| 0 | 1 - 126 | 0 | 1 - 62 | 0 | 1 - 30 | 0 | 1 - 14 | 0 | 1 - 6 | 0 | 1 - 2 |
| | | | | | | | | | | 4 | 5 - 6 |
| | | | | | | | | | | 8 | 9 - 10 |
| | | | | | | | | | | 12 | 13 - 14 |
| | | | | | | 16 | 17 - 30 | 16 | 17 - 22 | 16 | 17 - 18 |
| | | | | | | | | | | 20 | 21 - 22 |
| | | | | | | 24 | 25 - 30 | 24 | 25 - 26 | 24 | 25 - 26 |
| | | | | | | | | | | 28 | 29 - 30 |
| | | | | | | 32 | 33 - 46 | 32 | 33 - 38 | 32 | 33 - 34 |
| | | | | | | | | | | 36 | 37 - 38 |
| | | | | | | 40 | 41 - 46 | 40 | 41 - 42 | 40 | 41 - 42 |
| | | | | | | | | | | 44 | 45 - 46 |
| 64 | 65 - 126 | 64 | 65 - 94 | 64 | 65 - 78 | 48 | 49 - 54 | 48 | 49 - 50 | 48 | 49 - 50 |
| | | | | | | | | | | 52 | 53 - 54 |
| | | | | | | | | | | 56 | 57 - 58 |
| | | | | | | | | | | 60 | 61 - 62 |
| | | | | | | 64 | 65 - 70 | 64 | 65 - 66 | 64 | 65 - 66 |
| | | | | | | | | | | 68 | 69 - 70 |
| | | | | | | 72 | 73 - 78 | 72 | 73 - 74 | 72 | 73 - 74 |
| | | | | | | | | | | 76 | 77 - 78 |
| | | | | | | 80 | 81 - 86 | 80 | 81 - 82 | 80 | 81 - 82 |
| | | | | | | | | | | 84 | 85 - 86 |
| 128 | 129 - 254 | 128 | 129 - 190 | 96 | 97 - 126 | 88 | 89 - 94 | 88 | 89 - 90 | 88 | 89 - 90 |
| | | | | | | | | | | 92 | 93 - 94 |
| | | | | | | 96 | 97 - 110 | 96 | 97 - 102 | 96 | 97 - 98 |
| | | | | | | | | | | 100 | 101 - 102 |
| | | | | | | 104 | 105 - 110 | 104 | 105 - 106 | 104 | 105 - 106 |
| | | | | | | | | | | 108 | 109 - 110 |
| | | | | | | 112 | 113 - 118 | 112 | 113 - 114 | 112 | 113 - 114 |
| | | | | | | | | | | 116 | 117 - 118 |
| | | | | | | 120 | 121 - 126 | 120 | 121 - 122 | 120 | 121 - 122 |
| | | | | | | | | | | 124 | 125 - 126 |
| 192 | 193 - 254 | 192 | 193 - 222 | 128 | 129 - 158 | 128 | 129 - 142 | 128 | 129 - 134 | 128 | 129 - 130 |
| | | | | | | | | | | 132 | 133 - 134 |
| | | | | | | 136 | 137 - 142 | 136 | 137 - 138 | 136 | 137 - 138 |
| | | | | | | | | | | 140 | 141 - 142 |
| | | | | | | 144 | 145 - 158 | 144 | 145 - 146 | 144 | 145 - 146 |
| | | | | | | | | | | 148 | 149 - 150 |
| | | | | | | 152 | 153 - 158 | 152 | 153 - 154 | 152 | 153 - 154 |
| | | | | | | | | | | 156 | 157 - 158 |
| | | | | | | 160 | 161 - 166 | 160 | 161 - 162 | 160 | 161 - 162 |
| | | | | | | | | | | 164 | 165 - 166 |
| | | | | | | 168 | 169 - 174 | 168 | 169 - 170 | 168 | 169 - 170 |
| | | | | | | | | | | 172 | 173 - 174 |
| 224 | 225 - 254 | 224 | 225 - 222 | 160 | 161 - 190 | 176 | 177 - 182 | 176 | 177 - 178 | 176 | 177 - 178 |
| | | | | | | | | | | 180 | 181 - 182 |
| | | | | | | 184 | 185 - 190 | 184 | 185 - 186 | 184 | 185 - 186 |
| | | | | | | | | | | 188 | 189 - 190 |
| | | | | | | 192 | 193 - 198 | 192 | 193 - 194 | 192 | 193 - 194 |
| | | | | | | | | | | 196 | 197 - 198 |
| | | | | | | 200 | 201 - 206 | 200 | 201 - 202 | 200 | 201 - 202 |
| | | | | | | | | | | 204 | 205 - 206 |
| | | | | | | 208 | 209 - 214 | 208 | 209 - 210 | 208 | 209 - 210 |
| | | | | | | | | | | 212 | 213 - 214 |
| 240 | 241 - 254 | 240 | 241 - 222 | 224 | 225 - 238 | 216 | 217 - 222 | 216 | 217 - 218 | 216 | 217 - 218 |
| | | | | | | | | | | 220 | 221 - 222 |
| | | | | | | 224 | 225 - 230 | 224 | 225 - 226 | 224 | 225 - 226 |
| | | | | | | | | | | 228 | 229 - 230 |
| | | | | | | 232 | 233 - 238 | 232 | 233 - 234 | 232 | 233 - 234 |
| | | | | | | | | | | 236 | 237 - 238 |
| | | | | | | 240 | 241 - 246 | 240 | 241 - 242 | 240 | 241 - 242 |
| | | | | | | | | | | 244 | 245 - 246 |
| | | | | | | 248 | 249 - 254 | 248 | 249 - 250 | 248 | 249 - 250 |
| | | | | | | | | | | 252 | 253 - 254 |

Máscaras comunes

- /32: Se usa por ejemplo para publicar direcciones de loopback por medio de algún protocolo de ruteo dinámico. Solamente tiene un host, no tiene dirección de red ni de broadcast.
- /31: Tiene dos direcciones de host, no tiene dirección de red ni de broadcast. Es útil para enlaces punto a punto en WAN pero no muy usado. [Ver RFC-2021](#).
- /32: Tiene dos direcciones de host, una para la red y una para broadcast. Es lo más usado para enlaces punto a punto en WAN.

Fragmentación

Enviando PING con tamaño 6000 usando ping 192.168.1.6 -s 6000:

```
1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=3247) [Reassembled in #67]
1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=3247) [Reassembled in #67]
1514 Fragmented IP protocol (proto=ICMP 1, off=2960, ID=3247) [Reassembled in #67]
1514 Fragmented IP protocol (proto=ICMP 1, off=4440, ID=3247) [Reassembled in #67]
122 Echo (ping) reply id=0x095b, seq=13/3328, ttl=64 (request in 62)
```

Se partió en 5 fragmentos. Header IP de cada fragmento:

```
Internet Protocol Version 4, Src: 192.168.1.6, Dst: 192.168.1.16
Total Length: 1500
Identification: 0x3247 (12871)
Flags: 0x01 (More Fragments)
Fragment offset: 0
```

```
Internet Protocol Version 4, Src: 192.168.1.6, Dst: 192.168.1.16
Total Length: 1500
Identification: 0x3247 (12871)
Flags: 0x01 (More Fragments)
Fragment offset: 1480
```

```
Internet Protocol Version 4, Src: 192.168.1.6, Dst: 192.168.1.16
Total Length: 1500
Identification: 0x3247 (12871)
Flags: 0x01 (More Fragments)
Fragment offset: 2960
```

```
Internet Protocol Version 4, Src: 192.168.1.6, Dst: 192.168.1.16
Total Length: 1500
Identification: 0x3247 (12871)
Flags: 0x01 (More Fragments)
Fragment offset: 4440
```

```
Internet Protocol Version 4, Src: 192.168.1.6, Dst: 192.168.1.16
Total Length: 108
Identification: 0x3247 (12871)
Flags: 0x00
Fragment offset: 5920
```

Al hacer ping grande sin fragmentar me dice error en la terminal en donde hago ping con ping 192.168.1.6 -s 6000 -M do:

```
ping: local error: Message too long, mtu=1500
```

Traceroute

Al hacer sudo traceroute --icmp 8.8.8.8:

```
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
1 192.168.66.1 (192.168.66.1) 0.242 ms 0.277 ms 0.328 ms
```

```

2 * * *
3 gw.unrc.edu.ar (200.7.141.2) 0.871 ms 0.894 ms 0.901 ms
4 10.7.1.1 (10.7.1.1) 0.943 ms 1.588 ms 1.607 ms
5 host173.181-15-2.telecom.net.ar (181.15.2.173) 16.401 ms 16.408 ms 16.412 ms
6 host232.181-88-65.telecom.net.ar (181.88.65.232) 16.338 ms 16.390 ms 16.341 ms
7 host217.181-88-145.telecom.net.ar (181.88.145.217) 24.711 ms 24.737 ms 24.737 ms
8 host158.181-88-80.telecom.net.ar (181.88.80.158) 32.513 ms 32.536 ms 32.541 ms
9 host114.190-224-165.telecom.net.ar (190.224.165.114) 26.148 ms 28.284 ms 28.302 ms
10 72.14.217.180 (72.14.217.180) 26.516 ms 26.539 ms 26.419 ms
11 108.170.248.241 (108.170.248.241) 26.386 ms 26.616 ms 26.628 ms
12 108.170.227.7 (108.170.227.7) 31.107 ms 29.525 ms 29.550 ms
13 google-public-dns-a.google.com (8.8.8.8) 24.699 ms 24.477 ms 24.538 ms

192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=1/256, ttl=1 (no
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=2/512, ttl=1 (no
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=3/768, ttl=1 (no
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=4/1024, ttl=2 (no
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=5/1280, ttl=2 (no
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=6/1536, ttl=2 (no
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=7/1792, ttl=3 (no
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=8/2048, ttl=3 (no
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=9/2304, ttl=3 (no
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=10/2560, ttl=4 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=11/2816, ttl=4 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=12/3072, ttl=4 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=13/3328, ttl=5 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=14/3584, ttl=5 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=15/3840, ttl=5 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=16/4096, ttl=6 (r
192.168.66.1 192.168.1.16 ICMP 102 Time-to-live exceeded (Time to live exceeded in tra
192.168.66.1 192.168.1.16 ICMP 102 Time-to-live exceeded (Time to live exceeded in tra
192.168.66.1 192.168.1.16 ICMP 102 Time-to-live exceeded (Time to live exceeded in tra
200.7.141.2 192.168.1.16 ICMP 102 Time-to-live exceeded (Time to live exceeded in tra
200.7.141.2 192.168.1.16 ICMP 102 Time-to-live exceeded (Time to live exceeded in tra
200.7.141.2 192.168.1.16 ICMP 102 Time-to-live exceeded (Time to live exceeded in tra
10.7.1.1 192.168.1.16 ICMP 102 Time-to-live exceeded (Time to live exceeded in tra
10.7.1.1 192.168.1.16 ICMP 102 Time-to-live exceeded (Time to live exceeded in tra
10.7.1.1 192.168.1.16 ICMP 102 Time-to-live exceeded (Time to live exceeded in tra
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=17/4352, ttl=6 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=18/4608, ttl=6 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=19/4864, ttl=7 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=20/5120, ttl=7 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=21/5376, ttl=7 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=22/5632, ttl=8 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=23/5888, ttl=8 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=24/6144, ttl=8 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=25/6400, ttl=9 (r
181.88.65.232 192.168.1.16 ICMP 110 Time-to-live exceeded (Time to live exceeded in tra
181.15.2.173 192.168.1.16 ICMP 110 Time-to-live exceeded (Time to live exceeded in tra
181.15.2.173 192.168.1.16 ICMP 110 Time-to-live exceeded (Time to live exceeded in tra
181.15.2.173 192.168.1.16 ICMP 110 Time-to-live exceeded (Time to live exceeded in tra
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=26/6656, ttl=9 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=27/6912, ttl=9 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=28/7168, ttl=10 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=29/7424, ttl=10 (r
181.88.65.232 192.168.1.16 ICMP 110 Time-to-live exceeded (Time to live exceeded in tra
181.88.65.232 192.168.1.16 ICMP 110 Time-to-live exceeded (Time to live exceeded in tra
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=30/7680, ttl=10 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=31/7936, ttl=11 (r
192.168.1.16 8.8.8.8 ICMP 74 Echo (ping) request id=0x57fa, seq=32/8192, ttl=11 (r

```

| | | | |
|-----------------|--------------|----------|---|
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=33/8448, ttl=11 |
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=34/8704, ttl=12 |
| 181.88.145.217 | 192.168.1.16 | ICMP 110 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 181.88.145.217 | 192.168.1.16 | ICMP 110 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 181.88.145.217 | 192.168.1.16 | ICMP 110 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=35/8960, ttl=12 |
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=36/9216, ttl=12 |
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=37/9472, ttl=13 |
| 190.224.165.114 | 192.168.1.16 | ICMP 110 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=38/9728, ttl=13 |
| 181.88.80.158 | 192.168.1.16 | ICMP 110 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 181.88.80.158 | 192.168.1.16 | ICMP 110 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 181.88.80.158 | 192.168.1.16 | ICMP 110 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=39/9984, ttl=13 |
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=40/10240, ttl=14 |
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=41/10496, ttl=14 |
| 72.14.217.180 | 192.168.1.16 | ICMP 70 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 72.14.217.180 | 192.168.1.16 | ICMP 70 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=42/10752, ttl=14 |
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=43/11008, ttl=15 |
| 190.224.165.114 | 192.168.1.16 | ICMP 110 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 190.224.165.114 | 192.168.1.16 | ICMP 110 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 108.170.248.241 | 192.168.1.16 | ICMP 102 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 72.14.217.180 | 192.168.1.16 | ICMP 70 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=44/11264, ttl=15 |
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=45/11520, ttl=15 |
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=46/11776, ttl=16 |
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=47/12032, ttl=16 |
| 108.170.248.241 | 192.168.1.16 | ICMP 102 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 108.170.248.241 | 192.168.1.16 | ICMP 102 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=48/12288, ttl=16 |
| 192.168.1.16 | 8.8.8.8 | ICMP 74 | Echo (ping) request id=0x57fa, seq=49/12544, ttl=17 |
| 8.8.8.8 | 192.168.1.16 | ICMP 74 | Echo (ping) reply id=0x57fa, seq=37/9472, ttl=51 |
| 8.8.8.8 | 192.168.1.16 | ICMP 74 | Echo (ping) reply id=0x57fa, seq=38/9728, ttl=51 |
| 108.170.227.7 | 192.168.1.16 | ICMP 70 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 108.170.227.7 | 192.168.1.16 | ICMP 70 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 108.170.227.7 | 192.168.1.16 | ICMP 70 | Time-to-live exceeded (Time to live exceeded in transmission) |
| 8.8.8.8 | 192.168.1.16 | ICMP 74 | Echo (ping) reply id=0x57fa, seq=39/9984, ttl=51 |
| 8.8.8.8 | 192.168.1.16 | ICMP 74 | Echo (ping) reply id=0x57fa, seq=40/10240, ttl=51 |
| 8.8.8.8 | 192.168.1.16 | ICMP 74 | Echo (ping) reply id=0x57fa, seq=41/10496, ttl=51 |
| 8.8.8.8 | 192.168.1.16 | ICMP 74 | Echo (ping) reply id=0x57fa, seq=42/10752, ttl=51 |
| 8.8.8.8 | 192.168.1.16 | ICMP 74 | Echo (ping) reply id=0x57fa, seq=43/11008, ttl=51 |
| 8.8.8.8 | 192.168.1.16 | ICMP 74 | Echo (ping) reply id=0x57fa, seq=44/11264, ttl=51 |
| 8.8.8.8 | 192.168.1.16 | ICMP 74 | Echo (ping) reply id=0x57fa, seq=45/11520, ttl=51 |
| 8.8.8.8 | 192.168.1.16 | ICMP 74 | Echo (ping) reply id=0x57fa, seq=46/11776, ttl=51 |
| 8.8.8.8 | 192.168.1.16 | ICMP 74 | Echo (ping) reply id=0x57fa, seq=47/12032, ttl=51 |
| 8.8.8.8 | 192.168.1.16 | ICMP 74 | Echo (ping) reply id=0x57fa, seq=48/12288, ttl=51 |
| 8.8.8.8 | 192.168.1.16 | ICMP 74 | Echo (ping) reply id=0x57fa, seq=49/12544, ttl=51 |

ICMPv4

Internet Control Message Protocol

- Va sobre IP.
- Se usa para diagnóstico y control de IP.

Header

- Type (1 byte): Tipo de mensaje.

- Code (1 byte): Es como un subtipo.
- Checksum (2 bytes): Checksum de todo el header.
- Resto del header (4 bytes): Los contenidos dependen del tipo de mensaje.
- Datos: Son cualquier cosa.

Ping

- Type: 8 en request, 0 en reply.
- Code: 0.
- Resto del header:
 - Identifier (2 bytes): Identifica el proceso que envía los pings.
 - Sequence number (2 bytes): Número de ping enviado.
- Datos: Puede ser cualquier cosa, suele llevar timestamp.

Otros usos

Hay muchos más, pongo los más comunes junto a sus valores (Type, Code):

- Cuando se termina el TTL se envía un «ICMP Time Exceeded» (11, 0).
- Red de destino inalcanzable (3, 0).
- Host de destino inalcanzable (3, 1).
- Protocolo de destino inalcanzable (3, 2).
- Puerto de destino inalcanzable (3, 3).

Redirección ICMP

Por hacer

Buscar sobre ICMP Redirect. Creo que un router informa a un host de una ruta más corta

Capturas

Echo Request:

```
Frame 11: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on
           interface 0
Ethernet II, Src: HonHaiPr_13:7f:55 (08:3e:8e:13:7f:55), Dst:
           Tp-LinkT_22:9a:f2 (ec:08:6b:22:9a:f2)
Internet Protocol Version 4, Src: 192.168.0.20, Dst: 8.8.8.8
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 84
Identification: 0xcdbe (52670)
Flags: 0x02 (Don't Fragment)
Fragment offset: 0
Time to live: 64
Protocol: ICMP (1)
Header checksum: 0x9c1e [validation disabled]
[Header checksum status: Unverified]
Source: 192.168.0.20
```

```

Destination: 8.8.8.8
[Source GeoIP: Unknown]
[Destination GeoIP: United States, AS15169 Google Inc., Mountain View, CA,
 37.386002, -122.083801]
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x5b79 [correct]
[Checksum Status: Good]
Identifier (BE): 14306 (0x37e2)
Identifier (LE): 57911 (0xe237)
Sequence number (BE): 1 (0x0001)
Sequence number (LE): 256 (0x0100)
[Response frame: 12]
Timestamp from icmp data: Jun 16, 2018 20:08:22.000000000 -03
[Timestamp from icmp data (relative): 0.449866492 seconds]
Data (48 bytes)
  Data: 13dd060000000000101112131415161718191a1b1c1d1e1f...
  [Length: 48]

```

Echo Reply:

```

Frame 12: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on
  interface 0
Ethernet II, Src: Tp-LinkT_22:9a:f2 (ec:08:6b:22:9a:f2), Dst:
  HonHaiPr_13:7f:55 (08:3e:8e:13:7f:55)
Internet Protocol Version 4, Src: 8.8.8.8, Dst: 192.168.0.20
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 84
  Identification: 0x2f2b (12075)
  Flags: 0x00
  Fragment offset: 0
  Time to live: 50
  Protocol: ICMP (1)
  Header checksum: 0x88b2 [validation disabled]
  [Header checksum status: Unverified]
  Source: 8.8.8.8
  Destination: 192.168.0.20
  [Source GeoIP: United States, AS15169 Google Inc., Mountain View, CA,
   37.386002, -122.083801]
  [Destination GeoIP: Unknown]
Internet Control Message Protocol
Type: 0 (Echo (ping) reply)
Code: 0
Checksum: 0x6379 [correct]
[Checksum Status: Good]
Identifier (BE): 14306 (0x37e2)
Identifier (LE): 57911 (0xe237)
Sequence number (BE): 1 (0x0001)
Sequence number (LE): 256 (0x0100)
[Request frame: 11]
[Response time: 24.340 ms]
Timestamp from icmp data: Jun 16, 2018 20:08:22.000000000 -03
[Timestamp from icmp data (relative): 0.474206330 seconds]
Data (48 bytes)
  Data: 13dd060000000000101112131415161718191a1b1c1d1e1f...
  [Length: 48]

```

NAT

Network Address Translation.

Basic NAT

- Hace un mapeo de direcciones IP uno a uno.

NAPT

El tipo de NAT más usado es NAPT Dinámico (Network Address Port Translation):

- Hace IP masquerading o One-to-many NAT: es una técnica que esconde un espacio de direcciones IP (privadas) detrás de una sola IP pública.
- Se cambia la IP de origen de los paquetes que salen del Router.
- Hace falta porque las IP privadas no son ruteables en la internet.
- Para permitir que salgan varios usuarios se debe cambiar también el puerto de origen TCP o UDP, para de esa forma identificar las conexiones a distintos usuarios.
- En el caso de ping ICMP se cambia el Identifier.

Tipos

- Según los campos que modifican:
 - NAT Básico: Sólo se cambia la dirección IP, requiere una IP pública por cada IP privada en uso.
 - NAPT (Network Address Port Translation): Modifica la dirección IP y además el puerto TCP o UDP. Permite sacar muchas IP privadas por un número menor de direcciones públicas.
- Según si es estático o dinámico:
 - Estático: La tabla de correspondencias es estáticas.
 - Dinámico: La tabla se va modificando, permite reutilizar direcciones de salida.

Otras terminologías

- DNAT (Destination Network Address Translation): Cambia la dirección IP de destino.
- SNAT: Puede significar:
 - Source NAT: Se cambia la IP de origen, lo común.
 - Stateful NAT, Secure NAT, que se yo.

DHCP

Dynamic Host Configuration Protocol.

- Viene por UDP
 - Puerto 68 en cliente
 - Puerto 67 en servidor
- Es la evolución del protocolo BOOTP.

Header

- Message type (OP) (1 byte): Indica petición (1) o respuesta (2).
- Hardware type (HTYPE) (1 byte): Protocolo de enlace, si es Ethernet es 1.

- Hardware address length (HLEN) (1 byte): Longitud de la dirección de enlace en bytes, si es Ethernet es 6.
- Hops (HOPS) (1 byte): Empieza en cero, cada gateway incrementa el salto, si llega a 3 se supone que hubo un bucle.
- Transaction ID (XID) (4 bytes): Número aleatorio que identifica a esta transacción (DHCPDISCOVER, DHCPOFFER, DHCPREQUEST y DHCPACK).
- Seconds elapsed (SECS) (2 bytes): Fijado por el cliente, segundos desde que el cliente arrancó.
- BOOTP flags (FLAGS) (2 bytes): Solamente se usa el primer bit para indicar si el cliente quiere que el servidor responda como broadcast (1) o unicast (0). Se usa siempre unicast salvo en clientes que no puedan recibir unicast antes de inicializar su stack TCP/IP.
- Client IP (CIADDR) (4 bytes): Generalmente es 0.0.0.0 salvo por ejemplo cuando se hace una renovación.
- Your IP (YIADDR) (4 bytes): Fijado por el servidor si CIADDR era 0.0.0.0.
- Server IP (SIADDR) (4 bytes): IP del servidor, fijada por el servidor.
- Relay IP (GIADDR) (4 bytes): Fijado por el relay si es que se está usando uno.
- Client MAC (CHADDR) (16 bytes): Fijada por el cliente, como sobra espacio tiene padding de ceros.
- Server hostname (SNAME) (64 bytes): Es opcional, pero si no se usa queda el espacio vacío con ceros.
- Boot filename (FILE) (128 bytes): En DHCPDISCOVER no tiene nada, en DHCPOFFER el servidor fija un nombre de archivo.
- Opciones (OPTIONS) (variable): Ver abajo

Opciones

Hay muchísimas, pongo las mas usadas:

- Generales, se usan siempre:
 - DHCP Message type (53): Indica el tipo de mensaje, por ejemplo:
 - DHCPDISCOVER: 1
 - DHCPOFFER: 2
 - DHCPREQUEST: 3
 - DHCPACK: 5
 - Endmark (255): Indica fin de opciones.
- Usadas en DHCPDISCOVER y DHCPREQUEST.
 - Parameter request list (55): Lista de parámetros que requiere el cliente. Son varios números que se corresponden a las opciones que explico abajo.
- Usadas en DHCPOFFER y DHCPACK.
 - Subnet mask (1): Máscara de red a usar por el cliente
 - Router (3): Gateway a usar por el cliente.
 - IP address lease time (51): Tiempo de lease para la IP dada.
 - DNS servers (8): Servidores DNS para el cliente.
 - DHCP server identifier (54): IP del servidor DHCP.
- Usadas en DHCPREQUEST:
 - Requested IP address (50): IP que el cliente está solicitando.

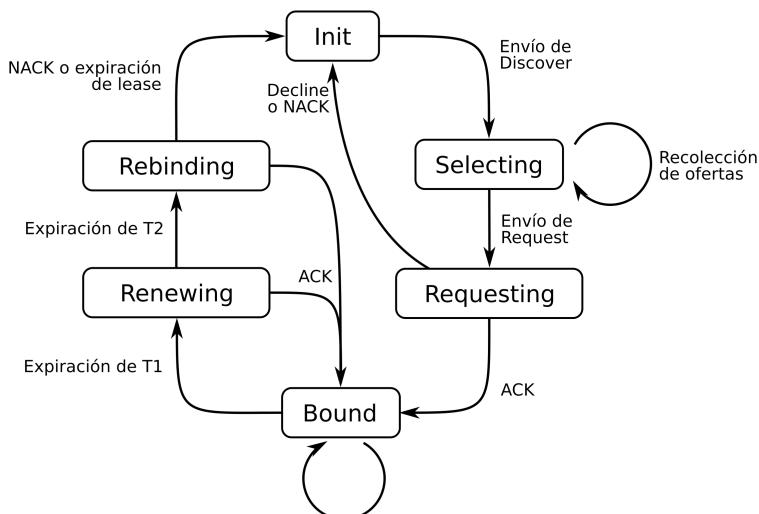
Funcionamiento

Los datagramas más importantes son DHCPDISCOVER, DHCPOFFER, DHCPREQUEST y DHCPACK:

- Descubrimiento (DHCPDISCOVER): Cliente pregunta que DHCPs hay.
 - Origen:
 - MAC: MAC Cliente.
 - IP: 0.0.0.0.
 - Puerto: 68.
 - Destino:
 - MAC: FF:FF:FF:FF:FF:FF.
 - IP: 255.255.255.255.
 - Puerto: 67
 - DHCP:
 - YIADDR: 0.0.0.0.
 - DHCP message type: 1 (Discover).
 - Lista de parámetros requeridos.
- Oferta (DHCPOFFER): Servidores DHCP reservan una IP y la ofrecen, indicando además el tiempo de lease (arrendamiento).
 - Origen:
 - MAC: MAC Servidor.
 - IP: IP Servidor.
 - Puerto: 67.
 - Destino:
 - MAC: MAC Cliente
 - IP: IP propuesta para Cliente.
 - Puerto: 68
 - DHCP:
 - YIADDR: IP propuesta para cliente.
 - DHCP message type: 2 (Offer).
 - Tiempo de lease, máscara de red, gateway, DNS.

- Solicitud (DHCPREQUEST): Cliente pide a un servidor DHCP esa IP, es broadcast para avisarle a los potenciales otros servidores que no los elegi.
 - Origen:
 - MAC: MAC Cliente.
 - IP: 0.0.0.0.
 - Puerto: 68.
 - Destino:
 - MAC: FF:FF:FF:FF:FF:FF.
 - IP: 255.255.255.255.
 - Puerto: 67
 - DHCP:
 - YIADDR: 0.0.0.0.
 - ~~DHCP message type: 2 (Request).~~
 - IP requerida por el cliente.
- ACK (DHCPACK): Servidor confirma.
 - Origen:
 - MAC: MAC Servidor.
 - IP: IP Servidor.
 - Puerto: 67.
 - Destino:
 - MAC: MAC Cliente
 - IP: IP Cliente.
 - Puerto: 68
 - DHCP:
 - YIADDR: IP propuesta para cliente.
 - DHCP message type: 5 (ACK).
 - Tiempo de lease, máscara de red, gateway, DNS.

Estados



- Initialization
- Selection: Espera todas las ofertas de los servidores
- Request: Pide a un servidor y espera respuesta.
- Bound: Una vez que le llegó la confirmación del servidor. En este momento la PC trabaja normalmente.
- Renew: Envía un DHCPREQUEST para renovar (al 50% del tiempo) y espera respuesta.
- Rebind: Envía un DHCPREQUEST para renovar (al finalizar el tiempo) y espera respuesta.

DHCP relaying

Cuando hay varias subredes, puede haber un servidor DHCP por subred o puede haber un servidor que sirva a varias redes. El problema es que en este caso el servidor y el cliente están en redes distintas, el cliente no tiene una IP ruteable y tampoco sabe la IP del servidor DHCP.

Para permitir a los clientes comunicarse con los servidores, se instalan *DHCP relay agents*. Estos agentes de retransmisión cuando reciben un broadcast DHCP reenvían el mensaje a uno o varios servidores DHCP presentes en otras redes mediante unicast.

Hay un campo específico en la trama llamado GIADDR, en donde el gateway coloca su IP, de esta forma el servidor DHCP sabe que debe responder a esa IP en particular y que debe reservarle una IP al cliente en esa red.

Para la comunicación entre relay y servidor generalmente se usa el puerto 67 para origen y destino.

Capturas

Discover, UDP 68 -> 67:

```
Frame 55: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on
           interface 0
Ethernet II, Src: HonHaiPr_13:7f:55 (08:3e:8e:13:7f:55), Dst: Broadcast
           (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
 0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x10 (DSCP: Unknown, ECN: Not-ECT)
  Total Length: 328
  Identification: 0x0000 (0)
  Flags: 0x00
  Fragment offset: 0
  Time to live: 128
  Protocol: UDP (17)
  Header checksum: 0x3996 [validation disabled]
  [Header checksum status: Unverified]
  Source: 0.0.0.0
  Destination: 255.255.255.255
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
User Datagram Protocol, Src Port: 68, Dst Port: 67
  Source Port: 68
  Destination Port: 67
  Length: 308
  Checksum: 0xbff5 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 0]
Bootstrap Protocol (Discover)
  Message type: Boot Request (1)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
```

```

Transaction ID: 0xa1368e3d
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
    0... .... .... = Broadcast flag: Unicast
    .000 0000 0000 0000 = Reserved flags: 0x0000
Client IP address: 0.0.0.0
Your (client) IP address: 0.0.0.0
Next server IP address: 0.0.0.0
Relay agent IP address: 0.0.0.0
Client MAC address: HonHaiPr_13:7f:55 (08:3e:8e:13:7f:55)
Client hardware address padding: 00000000000000000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
Option: (53) DHCP Message Type (Discover)
    Length: 1
    DHCP: Discover (1)
Option: (12) Host Name
    Length: 16
    Host Name: mbernardi-laptop
Option: (55) Parameter Request List
    Length: 13
    Parameter Request List Item: (1) Subnet Mask
    Parameter Request List Item: (28) Broadcast Address
    Parameter Request List Item: (2) Time Offset
    Parameter Request List Item: (3) Router
    Parameter Request List Item: (15) Domain Name
    Parameter Request List Item: (6) Domain Name Server
    Parameter Request List Item: (119) Domain Search
    Parameter Request List Item: (12) Host Name
    Parameter Request List Item: (44) NetBIOS over TCP/IP Name Server
    Parameter Request List Item: (47) NetBIOS over TCP/IP Scope
    Parameter Request List Item: (26) Interface MTU
    Parameter Request List Item: (121) Classless Static Route
    Parameter Request List Item: (42) Network Time Protocol Servers
Option: (255) End
    Option End: 255
Padding: 000000000000000000000000000000000000000000000000000000000000000

```

Offer, UDP 67 -> 68:

```

Frame 56: 590 bytes on wire (4720 bits), 590 bytes captured (4720 bits) on
interface 0
Ethernet II, Src: Tp-LinkT_c7:77:b4 (70:4f:57:c7:77:b4), Dst:
    HonHaiPr_13:7f:55 (08:3e:8e:13:7f:55)
Internet Protocol Version 4, Src: 192.168.2.1, Dst: 192.168.2.101
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 576
Identification: 0xb762 (46946)
Flags: 0x00
Fragment offset: 0
Time to live: 64
Protocol: UDP (17)
Header checksum: 0x3b94 [validation disabled]
[Header checksum status: Unverified]
Source: 192.168.2.1
Destination: 192.168.2.101
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]

```

```
User Datagram Protocol, Src Port: 67, Dst Port: 68
  Source Port: 67
  Destination Port: 68
  Length: 556
  Checksum: 0xcb8a [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
Bootstrap Protocol (Offer)
  Message type: Boot Reply (2)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0xa1368e3d
  Seconds elapsed: 0
  Bootp flags: 0x0000 (Unicast)
    0... .... .... = Broadcast flag: Unicast
    .000 0000 0000 0000 = Reserved flags: 0x0000
  Client IP address: 0.0.0.0
  Your (client) IP address: 192.168.2.101
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address: HonHaiPr_13:7f:55 (08:3e:8e:13:7f:55)
  Client hardware address padding: 000000000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  Option: (53) DHCP Message Type (Offer)
    Length: 1
    DHCP: Offer (2)
  Option: (54) DHCP Server Identifier
    Length: 4
    DHCP Server Identifier: 192.168.2.1
  Option: (51) IP Address Lease Time
    Length: 4
    IP Address Lease Time: (7200s) 2 hours
  Option: (1) Subnet Mask
    Length: 4
    Subnet Mask: 255.255.255.0
  Option: (3) Router
    Length: 4
    Router: 192.168.2.1
  Option: (6) Domain Name Server
    Length: 8
    Domain Name Server: 200.7.141.6
    Domain Name Server: 200.7.141.7
  Option: (255) End
    Option End: 255
  Padding: 000000000000000000000000000000000000000000000000000000000000000...
```

Request, 68 -> 67:

```
Frame 57: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on
  interface 0
Ethernet II, Src: HonHaiPr_13:7f:55 (08:3e:8e:13:7f:55), Dst: Broadcast
  (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x10 (DSCP: Unknown, ECN: Not-ECT)
  Total Length: 328
  Identification: 0x0000 (0)
```

```

Flags: 0x00
Fragment offset: 0
Time to live: 128
Protocol: UDP (17)
Header checksum: 0x3996 [validation disabled]
[Header checksum status: Unverified]
Source: 0.0.0.0
Destination: 255.255.255.255
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
User Datagram Protocol, Src Port: 68, Dst Port: 67
Source Port: 68
Destination Port: 67
Length: 308
Checksum: 0xfd67 [unverified]
[Checksum Status: Unverified]
[Stream index: 0]
Bootstrap Protocol (Request)
Message type: Boot Request (1)
Hardware type: Ethernet (0x01)
Hardware address length: 6
Hops: 0
Transaction ID: 0xa1368e3d
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
    0... .... .... .... = Broadcast flag: Unicast
    .000 0000 0000 0000 = Reserved flags: 0x0000
Client IP address: 0.0.0.0
Your (client) IP address: 0.0.0.0
Next server IP address: 0.0.0.0
Relay agent IP address: 0.0.0.0
Client MAC address: HonHaiPr_13:7f:55 (08:3e:8e:13:7f:55)
Client hardware address padding: 000000000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
Option: (53) DHCP Message Type (Request)
    Length: 1
    DHCP: Request (3)
Option: (54) DHCP Server Identifier
    Length: 4
    DHCP Server Identifier: 192.168.2.1
Option: (50) Requested IP Address
    Length: 4
    Requested IP Address: 192.168.2.101
Option: (12) Host Name
    Length: 16
    Host Name: mbernardi-laptop
Option: (55) Parameter Request List
    Length: 13
    Parameter Request List Item: (1) Subnet Mask
    Parameter Request List Item: (28) Broadcast Address
    Parameter Request List Item: (2) Time Offset
    Parameter Request List Item: (3) Router
    Parameter Request List Item: (15) Domain Name
    Parameter Request List Item: (6) Domain Name Server
    Parameter Request List Item: (119) Domain Search
    Parameter Request List Item: (12) Host Name
    Parameter Request List Item: (44) NetBIOS over TCP/IP Name Server
    Parameter Request List Item: (47) NetBIOS over TCP/IP Scope

```

```

Parameter Request List Item: (26) Interface MTU
Parameter Request List Item: (121) Classless Static Route
Parameter Request List Item: (42) Network Time Protocol Servers
Option: (255) End
  Option End: 255
Padding: 00000000000000000000000000000000

```

ACK, UDP 67 -> 68:

```

Frame 58: 590 bytes on wire (4720 bits), 590 bytes captured (4720 bits) on
  interface 0
Ethernet II, Src: Tp-LinkT_c7:77:b4 (70:4f:57:c7:77:b4), Dst:
  HonHaiPr_13:7f:55 (08:3e:8e:13:7f:55)
Internet Protocol Version 4, Src: 192.168.2.1, Dst: 192.168.2.101
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSGP: CS0, ECN: Not-ECT)
Total Length: 576
Identification: 0xb762 (46946)
Flags: 0x00
Fragment offset: 0
Time to live: 64
Protocol: UDP (17)
Header checksum: 0x3b94 [validation disabled]
[Header checksum status: Unverified]
Source: 192.168.2.1
Destination: 192.168.2.101
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
User Datagram Protocol, Src Port: 67, Dst Port: 68
  Source Port: 67
  Destination Port: 68
  Length: 556
  Checksum: 0xc88a [unverified]
  [Checksum Status: Unverified]
  [Stream index: 1]
Bootstrap Protocol (ACK)
  Message type: Boot Reply (2)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0xa1368e3d
  Seconds elapsed: 0
  Bootp flags: 0x0000 (Unicast)
    0... .... .... .... = Broadcast flag: Unicast
    .000 0000 0000 0000 = Reserved flags: 0x0000
  Client IP address: 0.0.0.0
  Your (client) IP address: 192.168.2.101
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address: HonHaiPr_13:7f:55 (08:3e:8e:13:7f:55)
  Client hardware address padding: 000000000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  Option: (53) DHCP Message Type (ACK)
    Length: 1
    DHCP: ACK (5)
  Option: (54) DHCP Server Identifier
    Length: 4
    DHCP Server Identifier: 192.168.2.1

```

HDL C

High-Level Data Link Control

- Es un protocolo de capa 2 para WAN. Se usa en por ejemplo Serial.
 - Provee control de flujo y detección de errores.
 - Es Punto - Multipunto.
 - Está cayendo en desuso por PPP.

Control de flujo

Es una técnica para que el emisor no sobrecargue al receptor al enviarle más datos de los que pueda procesar. Vemos los protocolos ARQ, que según tengo entendido no son específicos a ningún protocolo de capa de enlace.

- Tiempos:
 - Tiempo de transmisión: Tiempo que se tarda en inyectar datos al medio.
 - Tiempo de propagación: Lo que tardan los datos en moverse por el medio.

Métodos

- On-Off:
 - No se usa. Creo que no es un protocolo ARQ.
 - La fuente envía mensajes sin esperar ACKs.
 - Cuando se está por llenar el buffer del receptor, éste envía una señal de stop que sería RNR (Receptor Not Ready).
 - Cuando vuelve a estar listo envía RR (Receptor Ready).
 - Stop and wait ARQ:
 - No se usa.
 - La fuente espera que le respondan un ACK antes de mandar la siguiente trama.
 - El destino detiene el flujo al no responder un ACK.
 - Es ineficiente
 - Ventanas deslizantes (Go Back N ARQ):
 - Al conectar define tamaño de ventana W.
 - Se transmiten W tramas numeradas sin esperar confirmación.

- El receptor tiene un buffer de tamaño W .
- El ACK dice cuál es la trama que espero que llegue. Se llama RR (Receptor Ready).
- El receptor puede dar un RNR (Receptor Not Ready).
- En full-duplex, al mandar información al mismo tiempo envía los RR.
- Hay un timeout.
- Si hay errores pido todas las tramas desde que ocurrió el error, REJ.
- Si hay una trama que no llegó, el receptor se da cuenta porque se saltó un número, entonces pido todas las tramas desde que ocurrió el error.
- Si la trama que no llegó es justo la última, el receptor nunca envía el ACK, entonces en el transmisor salta el timeout y pide un ACK al receptor para saber si se cayó la conexión y si llegó el último mensaje.
- Puede pasar que se pierda un ACK se pierda pero que el siguiente ACK llegue, en ese caso confirma todas las anteriores.
- Selective Repeat ARQ/Selective Reject ARQ:
 - No lo vimos.

Detección de errores

- Se usa CRC.
- Se debe decidir un polinomio P de largo k , el polinomio debe ser el mismo en el emisor y en el receptor.
- Para cada secuencia de datos genera $k-1$ bits de CRC. Para generar el CRC se agregan $k-1$ ceros al final de los datos y se divide por el polinomio P . El resto es el CRC, que se coloca en el lugar de los $k-1$ ceros que se agregaron.
- Al recibir se dividen los datos y el CRC concatenado por el mismo polinomio P , si el resto da cero está bien.

Tramas

- Tramas de información (I-frames): Transporte de datos, puede incluir información de control flujo y errores.
- Tramas de supervisión (S-frames): Control de flujo y errores.
- Tramas no numeradas (U-frames): Control de enlace, al inicio de la conexión generalmente.

Header

- Flag (8 bits): Delimitador de trama 01111110, *bit stuffing* si en los datos está este patrón.
- Address (8 bits o múltiplo de 8 bits): Identifica la estación secundaria, si son todos unos se trata de broadcast. Si el primer bit es un cero, significa que la dirección continúa en los siguientes 8 bits, se puede extender múltiples veces. Por culpa de ese bit, las direcciones tienen un múltiplo de 7 bits.
- Control (8 o 16 bits): Identifica tipo de trama.
- Tipo (16 bits): Sólo presente en *Cisco HDLC*, indica el protocolo encapsulado.
- Information (variable): Datos (IP). Sólo presente en tramas de información.
- FCS (16 o 32 bits): CRC.
- Flag (8 bits): El mismo delimitador de trama.

Control

Los tamaños en bits dependen del tipo de trama y de si se usa control de 8 o 16 bits, cuando se usan 16 bits las secuencias son de 6 bits en vez de 3 bits:

- Información (8 o 16 bits):
 - 0 (1 bit).
 - N(S) (3 o 6 bits): Secuencia de envío.
 - P/F (1 bit): Poll/Final bit.
 - N(R) (3 o 6 bits): Secuencia de recepción.
- Supervisión (8 o 16 bits):
 - 10 (2 bits).
 - S (2 bits): Opción de supervisión.
 - Padding (0 o 4 bits): Padding en el caso que se use control de 16 bits.
 - P/F (1 bit): Poll/Final bit.
 - N(R) (3 o 6 bits): Secuencia de recepción.
- No numeradas (8 bits):
 - 11 (2 bits).
 - S (2 bits): Opción no numerada
 - P/F (1 bit): Poll/Final bit.
 - M (3 bits): Opción no numerada.

El significado del Poll/Final bit depende del contexto:

- En tramas de órdenes solicita respuesta.
- En tramas de respuesta indica respuesta a una solicitud o final de transmisión.

Fases de operación

- Inicialización.
- Transferencia de datos.
- Desconexión.

Bit stuffing

Se necesita porque el delimitador de trama tiene 6 unos seguidos, y se necesita un sistema para impedir que esa secuencia ocurra en los datos.

Al enviar, si hay 5 unos seguidos se inserta un cero.

Al recibir, si recibe 5 unos seguidos mira el sexto bit:

- Si es un cero, se trata de datos, entonces borra el cero.
- Si es uno miro el séptimo bit:
 - Si es un cero se trata de un delimitador.
 - Si es un uno hubo algún error.

PPP

Point to Point Protocol.

- De WAN. Sucesor y variante de HDLC. Normalmente se usa en cables seriales, líneas de teléfono, etc. Para ADSL se usa PPPoE o PPPoA.
- Hace lo mismo que HDLC agregando autenticación, y en un ambiente orientado a la difusión.

Ventajas con HDLC:

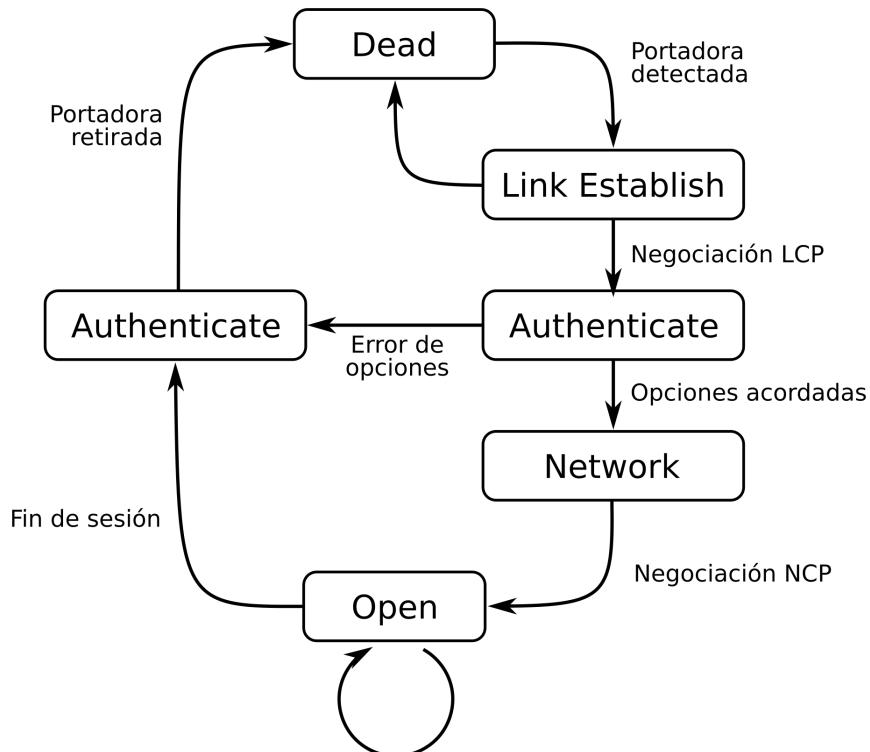
- Asignación dinámica de IP.
- Soporta varios protocolos de capa 3.
- Tiene mecanismo de control de red (NCP).
- Autenticación.
- Cifrado.
- Compresión de datos.

Header

Es un caso particular del Cisco HDLC.

- Flag (8 bits): Delimitador de trama 01111110, *bit stuffing* si en los datos está este patrón.
- Address (8 bits): Son siempre 11111111.
- Control (8 bits): Siempre 0x03 que significa *unnumbered data*.
- Protocol (16 bits): Protocolo encapsulado. Viene a ser el campo Type en Cisco HDLC pero ahora es estándar.
 - 0x0021 para IP.
 - 0xC021 para LCP.
 - 0x80XX para varios tipos de NCP.
- Information (variable): Datos (IP).
- FCS (16 bits): CRC.
- Flag (8 bits): El mismo delimitador de trama.

Estados



Parecido a HDLC, los estados son:

- Inicialización.
- Determinación de calidad de enlace (opcional).
- Transferencia de datos.
- Desconexión.

Control de enlace

Tiene un protocolo de control de enlace: LCP (Link Control Protocol).

- Controla la identidad del dispositivo cliente, aceptándolo o rechazándolo. Creo que a esta parte la hace PAP (más inseguro, contraseñas en texto plano) o CHAP (más seguro).
- Determina el tamaño aceptable de paquete.
- Busca errores de configuración.

Control de red

Tiene un mecanismo de control de red, que es medio de capa 3: NCP (Network Control Protocol). Para cada protocolo de red usado, hay un NCP correspondiente.

En el caso de IP, se usa IPCP (Internet Protocol Control Protocol), le va a dar la IP al router, no se usa DHCP porque es para LAN.

PPPoE

PPP over Ethernet.

- Va dentro de Ethernet.
- Se le agrega un encabezado PPPoE antes del encabezado PPP, aunque el encabezado PPP se recorta dejando solamente el campo del protocolo encapsulado, o sea que de PPP no queda nada basicamente.
- Cada extremo debe saber la MAC y número de sesión del otro, entonces hay un protocolo de descubrimiento.
- Puedo estar en estado de descubrimiento o de sesión.

Se usa para hacer un enlace punto a punto en redes LAN (que usan Ethernet) pero el uso más común es en las líneas de ADSL.

Como el par de cobre es punto a punto esperaría que usen PPP directamente. Eso se llama PPPoA (PPP over ATM), ATM es el protocolo que se usaba en ADSL, entonces el orden es: ADSL - ATM - PPP - IP.

Al usar PPPoE en ADSL (que más común), el orden es: ADSL - ATM - Ethernet - PPPoE - PPP - IP.

Estados

- Descubrimiento: Hay 4 mensajes:
 - PADI (PPPoE Active Discovery Initiation): El cliente pregunta por servidores.
 - PADO (PPPoE Active Discovery Offer): Un servidor ofrece.
 - PADR (PPPoE Active Discovery Request): El cliente acepta una propuesta.
 - PADS (PPPoE Active Discovery Session-confirmation): El servidor confirma.
- Sesión: Se envía el encabezado PPP en los datos junto con el paquete IP y todo lo demás. Los paquetes son unicast.

Apenas termina el establecimiento de PPPoE, se comienza a transmitir sobre PPP, específicamente los protocolos LCP (IPCP), PAP y CHAP.

Capturas

Sacadas de [esta captura](#).

PPPoE Active Discovery Initiation:

```
Frame 1: 24 bytes on wire (192 bits), 24 bytes captured (192 bits)
Ethernet II, Src: 20:28:18:a0:a9:d2 (20:28:18:a0:a9:d2),
    Dst: Broadcast (ff:ff:ff:ff:ff:ff)
PPP-over-Ethernet Discovery
0001 .... = Version: 1
.... 0001 = Type: 1
Code: Active Discovery Initiation (PADI) (0x09)
Session ID: 0x0000
Payload Length: 4
PPPoE Tags
```

PPPoE Active Discovery Offer:

```
Frame 2: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: Unispher_a4:10:be (00:90:1a:a4:10:be),
    Dst: 20:28:18:a0:a9:d2 (20:28:18:a0:a9:d2)
PPP-over-Ethernet Discovery
0001 .... = Version: 1
.... 0001 = Type: 1
Code: Active Discovery Offer (PAD0) (0x07)
Session ID: 0x0000
Payload Length: 35
PPPoE Tags
AC-Name: r-al121
AC-Cookie: bebcb53c10b32769a8661c36a45d8720
```

PPPoE Active Discovery Request:

```
Frame 3: 44 bytes on wire (352 bits), 44 bytes captured (352 bits)
Ethernet II, Src: 20:28:18:a0:a9:d2 (20:28:18:a0:a9:d2),
    Dst: Unispher_a4:10:be (00:90:1a:a4:10:be)
PPP-over-Ethernet Discovery
0001 .... = Version: 1
.... 0001 = Type: 1
Code: Active Discovery Request (PADR) (0x19)
Session ID: 0x0000
Payload Length: 24
PPPoE Tags
AC-Cookie: bebcb53c10b32769a8661c36a45d8720
```

PPPoE Active Discovery Session-confirmation:

```
Frame 4: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: Unispher_a4:10:be (00:90:1a:a4:10:be),
    Dst: 20:28:18:a0:a9:d2 (20:28:18:a0:a9:d2)
PPP-over-Ethernet Discovery
0001 .... = Version: 1
.... 0001 = Type: 1
Code: Active Discovery Session-confirmation (PADS) (0x65)
Session ID: 0x18b2
Payload Length: 4
PPPoE Tags
```

Lista completa con los demás mensajes:

| | | | |
|-----------|-----------|--------|--|
| MACclient | Broadcast | PPPoED | Active Discovery Initiation (PADI) |
| MACserver | MACclient | PPPoED | Active Discovery Offer (PAD0) AC-Name='serv' |
| MACclient | MACserver | PPPoED | Active Discovery Request (PADR) |
| MACserver | MACclient | PPPoED | Active Discovery Session-confirmation (PADS) |

| | | | |
|-----------|-----------|------------|--|
| MACclient | MACserver | PPP LCP | Configuration Request |
| MACserver | MACclient | PPP LCP | Configuration Request |
| MACserver | MACclient | PPP LCP | Configuration Ack |
| MACclient | MACserver | PPP LCP | Configuration Ack |
| MACclient | MACserver | PPP LCP | Echo Request |
| MACclient | MACserver | PPP PAP | Authenticate-Request (Peer-ID='usr', Password='pwd') |
| MACserver | MACclient | PPP LCP | Echo Reply |
| MACserver | MACclient | PPP PAP | Authenticate-Ack (Message='') |
| MACclient | MACserver | PPP IPCP | Configuration Request |
| MACclient | MACserver | PPP IPV6CP | Configuration Request |
| MACserver | MACclient | PPP IPCP | Configuration Nak |
| MACclient | MACserver | PPP IPCP | Configuration Request |
| MACserver | MACclient | PPP LCP | Protocol Reject |
| MACserver | MACclient | PPP IPCP | Configuration Ack |
| MACserver | MACclient | PPP IPCP | Configuration Request |
| MACclient | MACserver | PPP IPCP | Configuration Ack |
| MACclient | MACserver | PPP LCP | Echo Request |
| MACserver | MACclient | PPP LCP | Echo Reply |
| MACclient | MACserver | PPP LCP | Echo Request |
| MACserver | MACclient | PPP LCP | Echo Reply |
| MACclient | MACserver | PPP LCP | Echo Request |
| MACserver | MACclient | PPP LCP | Echo Reply |
| MACclient | MACserver | PPP LCP | Echo Request |
| MACserver | MACclient | PPP LCP | Echo Reply |

Header

- Version (4 bits): Siempre es 1.
- Type (4 bits): Siempre es 1.
- Code (8 bits):
 - 9: PADI.
 - 7: PADO.
 - 25: PADR.
 - 101: PADS.
 - 0: Estado de sesión, se envían paquetes IP.
- Session ID (16 bits): Identifica la sesión PPPoE.
- Length (16 bits): Largo de los datos en bytes.
- Datos (variable).

VLAN

- Separa dominios de difusión y dominios de colisión. Los dominios de colisión eran importantes cuando se usaban hubs, en el caso de los switches existen dominios de difusión que son separados por medio de VLAN.
- Los switches por defecto tienen todos los puertos en la VLAN 1, la 0 y la 255 no existen.
- La serie 3000 de Cisco tiene dos protocolos, al crear un trunk hay que elegir entre 802.1Q y ISL.

802.1Q

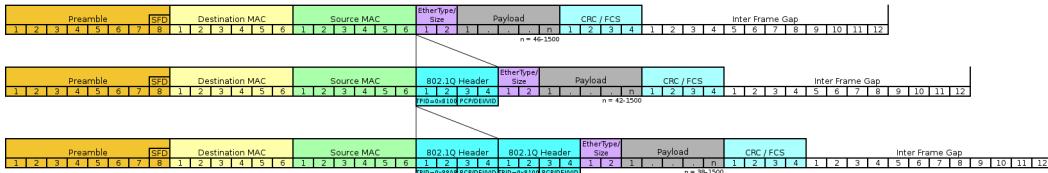
Se agrega un tag en el header Ethernet que tiene:

- Tag Protocol Identifier (TPID) (2 bytes): Tiene el valor 0x8100 para indicar que esta es una tag, ya que en paquetes normales esta posición llevaría el tipo de paquete Ethernet.

Teoría

- Priority Code Point (PCP) (3 bits): Se usa para marcar los paquetes con una cierta prioridad.
- Drop Eligible Indicator (DEI) (1 bit): Indica si este paquete se puede descartar en caso de congestión.
- VLAN Identifier (VID) (12 bits): Identificador de VLAN.

Se puede hacer *double tagging*, en tal caso hay dos tags uno al lado de otro.



By Luca Ghio CC BY-SA 4.0, from Wikimedia Commons.

ISL

Es un protocolo propietario de Cisco, encapsula a la trama Ethernet.

Tiene un header de 26 bytes, como datos lleva la trama Ethernet y finalmente tiene un CRC adicional de 4 bytes. No sé que pasa con el preámbulo de Ethernet, si queda adentro o afuera.

IPv6

- Tiene direcciones de 128 bits.
- Hay un encabezado obligatorio de tamaño fijo y luego encabezados opcionales.

Header

- Version (4 bits): Siempre es 6.
- Traffic Class (8 bits): Incluye los campos DS y ECN.
 - Differentiated Services (DS) (6 bits): Tipo de servicio, para QoS pero no se usa.
 - Explicit Congestion Notification (ECN) (2 bits): Se usa poco, indica que hay congestión en la red.
- Flow Label (20 bits): Para QoS.
- Payload Length (16 bits): Largo de los datos, incluyendo los encabezados de extensión.
- Next Header (8 bits): Tipo del próximo header, es el número de protocolo encapsulado, en el caso que se usen encabezados de extensión lleva el código que lo identifica.
- Hop Limit (8 bits): Este valor disminuye salto a salto y el paquete es descartado si llega a cero.
- Source address (128 bits): IP de origen.
- Destination address (128 bits): IP de destino.

Tipos de direcciones

- Según destino:
 - Unicast: Se entrega a una única interfaz.
 - Multicast: Se entrega a múltiples interfaces.
 - Anycast: Se entrega a cualquiera de las interfaces del conjunto.
 - No existe broadcast.
- Según rango:
 - Link-local: fe80::/16.

- Globalmente ruteable: 2000::/3.
- Global: 2001::/16.
- Documentación: 2001:db8::/32.
- Teredo: 2001:0::/32.
- 6to4: 2002::/16.
- Mapeada IPv4: ::ffff:xxxx/112.
- Compatible IPv4: ::xxxx/112 (Obsoleto).
- Direcciones especiales:
 - Loopback: ::1/128.
 - No especificada: ::/128.

SLAAC

Stateless Address Autoconfiguration.

- Permite a los hosts obtener direcciones IPv6 ruteables por medio de Router Advertisement y Router Solicitation (pertenecientes a ICMPv6).
- Si es necesario, se puede usar DHCPv6 en lugar de SLAAC.
- Primero el host obtiene su dirección link-local y envía un Router Solicitation. Luego el router responde con un Router Advertisement, indicando el prefijo de red que usará el host.

Por hacer

Agregar sobre como obtener parte de host: 64 bits EUI-64, a partir de MAC, aleatorio, dhcp o manual

Migración desde IPv4

El problema más común es que el núcleo de internet es IPv4, entonces hay que buscar la forma de enviar tráfico IPv6 a través de una red IPv4. Las soluciones que hay son:

- Túneles manuales: Se deben configurar manualmente en ambos extremos.
- Túneles automáticos: Por ejemplo 6to4.
- Traslación: Es una extensión de NAT para cambiar headers y direcciones.

Fragmentación

Se usa un header opcional para la fragmentación.

A diferencia de IPv4, sólo se puede hacer en los extremos. Los hosts intentan enviar los mensajes con un determinado tamaño, si se devuelve un *ICMP packet too big* se vuelve a intentar con el MTU dado en ese mensaje.

Enviando ping con un tamaño de 4000 usando ping6 fe80::2273:55ff:fe06:2451 -I wlan0 -s 4000:

```
fe80::39af:... fe80::2273:... IPv6      IPv6 fragment (off=0 more=y ident=0x79a0649e nxt=58)
fe80::39af:... fe80::2273:... IPv6      IPv6 fragment (off=1448 more=y ident=0x79a0649e nxt=59)
fe80::39af:... fe80::2273:... ICMPv6    Echo (ping) request id=0x3eeb, seq=4, hop limit=64 (n)
```

Se partió en 3 fragmentos, header de cada fragmento:

```
Internet Protocol Version 6, Src: fe80::39af:14fa:830b:c073, Dst: fe80::2273:55ff:fe06:2451
 0110 .... = Version: 6
  .... 0000 0000 .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
```

```

.... .... .... 1000 0010 0011 1011 0111 = Flow label: 0x823b7
Payload length: 1456
Next header: Fragment Header for IPv6 (44)
Hop limit: 64
Source: fe80::39af:14fa:830b:c073
Destination: fe80::2273:55ff:fe06:2451
[Destination SA MAC: ArrisGro_06:24:51 (20:73:55:06:24:51)]
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
Fragment Header for IPv6
    Next header: ICMPv6 (58)
    Reserved octet: 0x00
    0000 0000 0000 0... = Offset: 0 (0 bytes)
    .... .... .00. = Reserved bits: 0
    .... .... ....1 = More Fragments: Yes
    Identification: 0x79a0649e
Reassembled IPv6 in frame: 47

Frame 46: 1510 bytes on wire (12080 bits), 1510 bytes captured (12080 bits) on interface 0
Ethernet II, Src: HonHaiPr_13:7f:55 (08:3e:8e:13:7f:55), Dst: ArrisGro_06:24:51 (20:73:55:06:24:51)
Internet Protocol Version 6, Src: fe80::39af:14fa:830b:c073, Dst: fe80::2273:55ff:fe06:2451
    0110 .... = Version: 6
    .... 0000 0000 .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    .... .... .... 1000 0010 0011 1011 0111 = Flow label: 0x823b7
Payload length: 1456
Next header: Fragment Header for IPv6 (44)
Hop limit: 64
Source: fe80::39af:14fa:830b:c073
Destination: fe80::2273:55ff:fe06:2451
[Destination SA MAC: ArrisGro_06:24:51 (20:73:55:06:24:51)]
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
Fragment Header for IPv6
    Next header: ICMPv6 (58)
    Reserved octet: 0x00
    0000 0101 1010 1... = Offset: 181 (1448 bytes)
    .... .... .00. = Reserved bits: 0
    .... .... ....1 = More Fragments: Yes
    Identification: 0x79a0649e
Reassembled IPv6 in frame: 47

Frame 47: 1174 bytes on wire (9392 bits), 1174 bytes captured (9392 bits) on interface 0
Ethernet II, Src: HonHaiPr_13:7f:55 (08:3e:8e:13:7f:55), Dst: ArrisGro_06:24:51 (20:73:55:06:24:51)
Internet Protocol Version 6, Src: fe80::39af:14fa:830b:c073, Dst: fe80::2273:55ff:fe06:2451
    0110 .... = Version: 6
    .... 0000 0000 .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    .... .... .... 1000 0010 0011 1011 0111 = Flow label: 0x823b7
Payload length: 1120
Next header: Fragment Header for IPv6 (44)
Hop limit: 64
Source: fe80::39af:14fa:830b:c073
Destination: fe80::2273:55ff:fe06:2451
[Destination SA MAC: ArrisGro_06:24:51 (20:73:55:06:24:51)]
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
Fragment Header for IPv6
    Next header: ICMPv6 (58)
    Reserved octet: 0x00
    0000 1011 0101 0... = Offset: 362 (2896 bytes)
    .... .... .00. = Reserved bits: 0

```

....0 = More Fragments: No
Identification: 0x79a0649e

ICMPv6

Internet Control Message Protocol

- Va sobre IP.
- Se usa para diagnóstico y control de IP.

Header

- Type (1 byte): Tipo de mensaje.
- Code (1 byte): Es como un subtipo.
- Checksum (2 bytes): Checksum de todo el header pero también de un pseudo-header IPv6 para hacer checksum además a las direcciones IP.
- Resto del header (4 bytes): Los contenidos dependen del tipo de mensaje.
- Datos: Son cualquier cosa.

Ping

Header

- Type: 128 en request, 129 en reply.
- Code: 0 en ambos.
- Resto del header:
 - Identifier (2 bytes): Identifica el proceso que envía los pings.
 - Sequence number (2 bytes): Número de ping enviado.
- Datos: Puede ser cualquier cosa, suele llevar timestamp.

NDP

Neighbor Discovery Protocol.

- Permite encontrar vecinos y gateways, también cumple funciones similares a ARP.
- Define cinco mensajes ICMPv6:
 - Router Solicitation (133): Mensajes enviados por hosts para solicitar mensajes de Router Advertisement y así encontrar a Routers/Gateways.
 - Router Advertisement (134): Mensajes enviados periódicamente por routers para anunciar sus servicios y varios parámetros, permite aplicar SLAAC.
 - Neighbor Solicitation (135): Permite a los dispositivos determinar la dirección MAC de un vecino.
 - Neighbor Advertisement (136): Son las respuestas a Neighbor Solicitation.

Otros usos

Hay muchos más, pongo los más comunes junto a sus valores (Type, Code):

- Router Solicitation (133, 0).
- Router Advertisement (134, 0).
- Neighbor Solicitation (135, 0).

- Neighbor Advertisement (136, 0).
- Cuando se termina el TTL se envía un «Time Exceeded» (3, 0).
- No hay ruta a destino (1, 0).
- Dirección inalcanzable (3, 3).
- Puerto de destino inalcanzable (3, 4).

Capturas

Echo Request:

```
Frame 42: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
Ethernet II, Src: QuantaCo_43:f2:eb (08:9e:01:43:f2:eb), Dst: Cisco_bf:79:46 (00:15:62:bf:79:46)
Internet Protocol Version 6, Src: 2001:beba:baba:10:a9e:1ff:fe43:f2eb, Dst: 2001:beba:baba:10::2
  0110 .... = Version: 6
  .... 0000 0000 .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
  .... .... .... 0011 0011 0101 1010 0010 = Flow label: 0x335a2
  Payload length: 64
  Next header: ICMPv6 (58)
  Hop limit: 64
  Source: 2001:beba:baba:10:a9e:1ff:fe43:f2eb
  [Source SA MAC: QuantaCo_43:f2:eb (08:9e:01:43:f2:eb)]
  Destination: 2001:beba:baba:10::2
Internet Control Message Protocol v6
  Type: Echo (ping) request (128)
  Code: 0
  Checksum: 0xeab2 [correct]
  Identifier: 0x615b
  Sequence: 2
  Data (56 bytes)
```

Echo Reply:

```
Frame 43: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
Ethernet II, Src: Cisco_bf:79:46 (00:15:62:bf:79:46), Dst: QuantaCo_43:f2:eb (08:9e:01:43:f2:eb)
Internet Protocol Version 6, Src: 2001:beba:baba:10::2, Dst: 2001:beba:baba:10:a9e:1ff:fe43:f2eb
  0110 .... = Version: 6
  .... 0000 0000 .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
  .... .... .... 0011 0011 0101 1010 0010 = Flow label: 0x335a2
  Payload length: 64
  Next header: ICMPv6 (58)
  Hop limit: 64
  Source: 2001:beba:baba:10::2
  Destination: 2001:beba:baba:10:a9e:1ff:fe43:f2eb
  [Destination SA MAC: QuantaCo_43:f2:eb (08:9e:01:43:f2:eb)]
Internet Control Message Protocol v6
  Type: Echo (ping) reply (129)
  Code: 0
  Checksum: 0xe9b2 [correct]
  Identifier: 0x615b
  Sequence: 2
  [Response Time: 0.574 ms]
  Data (56 bytes)
```

Neighbor Solicitation:

```
Frame 37: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
Ethernet II, Src: QuantaCo_43:f2:eb (08:9e:01:43:f2:eb), Dst: IPv6mcast_ff:00:00:02 (33:33:33:33:33:33)
Internet Protocol Version 6, Src: 2001:beba:baba:10:a9e:1ff:fe43:f2eb, Dst: ff02::1:ff00:2
  0110 .... = Version: 6
  .... 0000 0000 .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
```

```

.... .... .... 0000 0000 0000 0000 0000 = Flow label: 0x000000
Payload length: 32
Next header: ICMPv6 (58)
Hop limit: 255
Source: 2001:beba:baba:10:a9e:1ff:fe43:f2eb
[Source SA MAC: QuantaCo_43:f2:eb (08:9e:01:43:f2:eb)]
Destination: ff02::1:ff00:2
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
Internet Control Message Protocol v6
Type: Neighbor Solicitation (135)
Code: 0
Checksum: 0x4bf4 [correct]
[Checksum Status: Good]
Reserved: 00000000
Target Address: 2001:beba:baba:10::2
ICMPv6 Option (Source link-layer address : 08:9e:01:43:f2:eb)

```

Neighbor Advertisement:

```

Frame 38: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
Ethernet II, Src: Cisco_bf:79:46 (00:15:62:bf:79:46), Dst: QuantaCo_43:f2:eb (08:9e:01:43:f2:eb)
Internet Protocol Version 6, Src: 2001:beba:baba:10::2, Dst: 2001:beba:baba:10:a9e:1ff:fe43:f2:eb
0110 .... = Version: 6
.... 1110 0000 .... .... .... .... = Traffic class: 0xe0 (DSCP: CS7, ECN: Not-ECT)
.... .... .... 0000 0000 0000 0000 0000 = Flow label: 0x000000
Payload length: 32
Next header: ICMPv6 (58)
Hop limit: 255
Source: 2001:beba:baba:10::2
Destination: 2001:beba:baba:10:a9e:1ff:fe43:f2eb
[Destination SA MAC: QuantaCo_43:f2:eb (08:9e:01:43:f2:eb)]
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
Internet Control Message Protocol v6
Type: Neighbor Advertisement (136)
Code: 0
Checksum: 0xef23 [correct]
[Checksum Status: Good]
Flags: 0xe0000000
Target Address: 2001:beba:baba:10::2
ICMPv6 Option (Target link-layer address : 00:15:62:bf:79:46)

```

Router Solicitation:

```

Frame 8: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
Ethernet II, Src: ce:96:8b:90:e5:bb (ce:96:8b:90:e5:bb), Dst: IPv6mcast_02 (33:33:00:00:00:00)
Internet Protocol Version 6, Src: fe80::cc96:8bff:fe90:e5bb, Dst: ff02::2
0110 .... = Version: 6
.... 0000 0000 .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
.... .... .... 0000 0000 0000 0000 0000 = Flow label: 0x000000
Payload length: 16
Next header: ICMPv6 (58)
Hop limit: 255
Source: fe80::cc96:8bff:fe90:e5bb
Destination: ff02::2
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
Internet Control Message Protocol v6
Type: Router Solicitation (133)
Code: 0

```

```

Checksum: 0xff67 [correct]
[Checksum Status: Good]
Reserved: 00000000
ICMPv6 Option (Source link-layer address : ce:96:8b:90:e5:bb)

```

Router Advertisement:

```

Frame 32: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
Ethernet II, Src: Cisco_bf:79:46 (00:15:62:bf:79:46), Dst: IPv6mcast_01 (33:33:00:00:00:01)
Internet Protocol Version 6, Src: fe80::215:62ff:febf:7946, Dst: ff02::1
    0110 .... = Version: 6
    .... 1110 0000 .... .... .... .... = Traffic class: 0xe0 (DSCP: CS7, ECN: Not-ECT)
    .... .... .... 0000 0000 0000 0000 = Flow label: 0x000000
    Payload length: 64
    Next header: ICMPv6 (58)
    Hop limit: 255
    Source: fe80::215:62ff:febf:7946
    [Source SA MAC: Cisco_bf:79:46 (00:15:62:bf:79:46)]
    Destination: ff02::1
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Internet Control Message Protocol v6
Type: Router Advertisement (134)
Code: 0
Checksum: 0xcae8 [correct]
[Checksum Status: Good]
Cur hop limit: 64
Flags: 0x00
Router lifetime (s): 1800
Reachable time (ms): 0
Retrans timer (ms): 0
ICMPv6 Option (Source link-layer address : 00:15:62:bf:79:46)
ICMPv6 Option (MTU : 1500)
ICMPv6 Option (Prefix information : 2001:beba:baba:10::/64)

```

6to4

- Permite transmitir paquetes IPv6 por una red IPv4 gracias a un router de borde.
- Es un mecanismo de transición para la migración de IPv4 a IPv6.
- No facilita la comunicación de hosts IPv4 con hosts IPv6, es para conectar hosts IPv6 con IPv6 a través de IPv4.
- Las direcciones comienzan con 2002::/16.
- Un router con 6to4 toma como decisión:
 - Si quiere llegar a una IPv6 dentro de la LAN sabe que está directamente conectada.
 - Si quiere llegar a una 2002::/16 sabe que tiene que mandar el paquete por la red IPv4, el destino IPv4 se determina de la dirección IPv6.
 - Si quiere llegar a alguna 2001::/16 tiene que enviar a algún router IPv6 que tenga acceso a la IPv6 nativa.
- Los paquetes IPv6 van dentro de los datos del paquete IPv4.
- Para permitir tráfico hacia redes IPv6 nativas (2001::/16) se establecieron los *Relay Routers*.
- Los *Relay routers* tienen, del lado IPv6, redes IPv6 nativas (2001::/16). Los routers de borde tienen redes con prefijo 2002::/16.
- Para que las IPv6 nativas envíen cosas a un IPv6 6to4, usan como gateway el *Relay router* más cercano, ya que éstos anuncian la ruta 2002::/16.

Direcciones

Hay un bloque de direcciones IPv6 para cada dirección IPv4:

2002:xxyy:wwzz::/48

En donde xxyy:wwzz es la dirección IPv4 en formato hexadecimal. Quedan 16 bits para hacer subredes

Por hacer

Hacer

<http://packetlife.net/blog/2010/mar/15/6to4-ipv6-tunneling/>

RIP

Routing Information Protocol.

Por hacer

Completar teoría.

Hay tres versiones de RIP: RIPv1, RIPv2 y RIPng. Las primeras dos son para IPv4, la tercera para IPV6.

- Sólo mira la cantidad de saltos.
- Cantidad máxima de saltos es 16.
- Cantidad máxima de redes es 25.
- Va sobre UDP 520.
- Se envían las rutas cada 30 segundos.
- Las rutas vencen 180 segundos después de caerse, para prevenir oscilaciones.
- RIPv1 no admite VLSM y es inseguro, porque no se identifica al que envía el paquete y un atacante podría redirigir todo por donde quiera.
- RIPv2 envía las máscaras de red y por lo tanto soporta VLSM, también autentica los routers.
- Conviene para redes chicas que no cambian mucho.
- Usa el algoritmo de vector distancia Bellman-Ford.
- RIPv1 anuncia por broadcast, RIPv2 anuncia por multicast.

Bellman-Ford

Es el algoritmo que usa RIP para determinar el mejor camino.

- Cada router tiene una tabla con las mejores distancias conocidas y los enlaces a utilizar para cada destino.
- Se intercambian las tablas de ruteo con los vecinos cada cierto tiempo.
- Una vez que el router tiene toda la información actualiza su tabla e informa a los vecinos de los cambios.
- A diferencia del algoritmo de Dijkstra:
 - Se intercambian tablas de ruteo en lugar de estados de enlace.
 - Se envía la información a sólo los vecinos en lugar de enviar a todos los routers.

- La convergencia es más lenta, hay mayor carga en la red pero se necesita una capacidad de procesamiento menor.

RIPng

Captura

En el GNS3 puse varios Mikrotik y varios Cisco. Después capturé los paquetes RIP en una de las conexiones.



En la foto se pueden ver los Cisco (routers azules), Mikrotik (routers azules y rojos) y el punto de captura (la lupa). En la captura se ven los paquetes RIP que se envían entre el Mikrotik encerrado con rojo (R3) y el Cisco encerrado en azul (R1), estos paquetes llevan las distancias entre el router que envía el paquete y una red determinada.

Los números en rojo son las distancias que informa el Mikrotik y los números en azul las distancias que informa el Cisco.

Paquete de Mikrotik a Cisco:

```
Frame 320: 266 bytes on wire (2128 bits), 266 bytes captured (2128 bits) on interface 0
Ethernet II, Src: 0c:45:23:ee:a3:00 (0c:45:23:ee:a3:00), Dst: IPv6mcast_09 (33:33:00:00:00:00)
Internet Protocol Version 6, Src: fe80::e45:23ff:feee:a300, Dst: ff02::9
User Datagram Protocol, Src Port: 521, Dst Port: 521
RIPng
  Command: Response (2)
  Version: 1
  Reserved: 0000
  Route Table Entry: IPv6 Prefix: 2001:1::/64 Metric: 16
  Route Table Entry: IPv6 Prefix: 2001:2::/64 Metric: 2
  Route Table Entry: IPv6 Prefix: 2001:3::/64 Metric: 1
  Route Table Entry: IPv6 Prefix: 2001:4::/64 Metric: 2
  Route Table Entry: IPv6 Prefix: 2001:5::/64 Metric: 3
  Route Table Entry: IPv6 Prefix: 2001:a::/64 Metric: 16
  Route Table Entry: IPv6 Prefix: 2001:b::/64 Metric: 16
  Route Table Entry: IPv6 Prefix: 2001:c::/64 Metric: 1
  Route Table Entry: IPv6 Prefix: 2001:d::/64 Metric: 1
  Route Table Entry: IPv6 Prefix: 2001:e::/64 Metric: 2
```

Paquete de Cisco a Mikrotik:

```
Frame 317: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits) on interface 0
Ethernet II, Src: c4:01:09:68:00:10 (c4:01:09:68:00:10), Dst: IPv6mcast_09 (33:33:00:00:00:00)
Internet Protocol Version 6, Src: fe80::c601:9ff:fe68:10, Dst: ff02::9
User Datagram Protocol, Src Port: 521, Dst Port: 521
RIPng
  Command: Response (2)
  Version: 1
  Reserved: 0000
  Route Table Entry: IPv6 Prefix: 2001:1::/64 Metric: 1
  Route Table Entry: IPv6 Prefix: 2001:a::/64 Metric: 1
  Route Table Entry: IPv6 Prefix: 2001:b::/64 Metric: 1
  Route Table Entry: IPv6 Prefix: 2001:e::/64 Metric: 2
```

```
Route Table Entry: IPv6 Prefix: 2001:2::/64 Metric: 2
Route Table Entry: IPv6 Prefix: 2001:5::/64 Metric: 3
```

Yo creo que el Mikrotik cuando informa las métricas pone la cantidad de saltos. Pero cuando sabe de el Cisco tiene una ruta más corta que a través de él, informa 16 que significa inalcanzable. El Cisco hace lo mismo pero en vez de poner métrica 16 directamente no informa de la ruta.

Al final la ruta del Cisco quedó:

```
IPv6 Routing Table - 15 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
      U - Per-user Static route
      I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
      O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
      ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

C 2001:1::/64 [0/0]
  via ::, FastEthernet0/0
L 2001:1::1/128 [0/0]
  via ::, FastEthernet0/0
R 2001:2::/64 [120/2]
  via FE80::C602:AFF:FE0D:1, FastEthernet0/1
R 2001:3::/64 [120/2]
  via FE80::E45:23FF:FE00:A300, FastEthernet1/0
R 2001:4::/64 [120/3]
  via FE80::E45:23FF:FE00:A300, FastEthernet1/0
R 2001:5::/64 [120/3]
  via FE80::C602:AFF:FE0D:1, FastEthernet0/1
C 2001:A::/64 [0/0]
  via ::, FastEthernet0/1
L 2001:A::1/128 [0/0]
  via ::, FastEthernet0/1
C 2001:B::/64 [0/0]
  via ::, FastEthernet1/0
L 2001:B::1/128 [0/0]
  via ::, FastEthernet1/0
R 2001:C::/64 [120/2]
  via FE80::C602:AFF:FE0D:1, FastEthernet0/1
  via FE80::E45:23FF:FE00:A300, FastEthernet1/0
R 2001:D::/64 [120/2]
  via FE80::E45:23FF:FE00:A300, FastEthernet1/0
R 2001:E::/64 [120/2]
  via FE80::C602:AFF:FE0D:1, FastEthernet0/1
L FE80::/10 [0/0]
  via ::, Null0
L FF00::/8 [0/0]
  via ::, Null0
```

Y la del Mikrotik:

```
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static,
r - rip, o - ospf, b - bgp, U - unreachable
```

| # | DST-ADDRESS | GATEWAY | DISTANCE |
|---|-----------------|-------------------------------|----------|
| 0 | ADr 2001:1::/64 | fe80::c601:9ff:fe68:1... {R1} | 120 |
| 1 | ADr 2001:2::/64 | fe80::c602:aff:fe0d:2... {R2} | 120 |
| 2 | ADC 2001:3::/64 | ether4 | 0 |
| 3 | ADr 2001:4::/64 | fe80::e45:23ff:fee7:3... {R4} | 120 |
| 4 | ADr 2001:5::/64 | fe80::c602:aff:fe0d:2... {R2} | 120 |
| 5 | ADr 2001:a::/64 | fe80::c601:9ff:fe68:1... {R1} | 120 |
| 6 | ADC 2001:b::/64 | ether1 | 0 |
| 7 | ADC 2001:c::/64 | ether2 | 0 |

| | | | | |
|---|-----|-------------|-------------------------------|-----|
| 8 | ADC | 2001:d::/64 | ether3 | 0 |
| 9 | ADr | 2001:e::/64 | fe80::c602:aff:fe0d:2... {R2} | 120 |

Revisé mirando el dibujo y todas las rutas están bien.

Algo raro es que en el caso del Cisco, para llegar a la red 2001:c::/64, es lo mismo usar como gateway al R2 que al R3, por lo tanto en la ruta están ambos gateway. El Mikrotik para llegar a la 2001:a::/64 también tiene dos opciones, R1 y R2, pero solamente tiene como gateway a R2.

OSPF

Open Shortest Path First.

- Se puede dividir en áreas, cada router sabe solamente los caminos dentro de su área, después sabe un resumen de las demás áreas pero nada más.
- Cada router tiene un ID de 24 bits, por defecto es la IP de una interfaz de loopback. De lo contrario se usa una IP de alguna interfaz pero puede traer problemas si esa interfaz se cae.
- Usa el algoritmo Dijkstra.
- La métrica depende de la implementación de cada dispositivo. Por ejemplo costo = $10^8 / \text{Ancho de banda en bits/s}$.
- El camino entre dos áreas siempre pasa por el área 0, salvo que se haga un camino virtual.
- OSPFv2 se usa para IPv4 mientras que OSPFv3 para IPv6.
- Utiliza multicast, va sobre IP.
- Intercambia información durante descubrimiento y cuando hay cambios. Cuando la red está estable sólo envía mensajes «Hello» para avisar que el enlace está activo.
- Se pueden definir áreas, para viajar entre dos áreas generalmente los paquetes pasan por el área 0 o backbone. Las demás áreas son *Stub* o de tránsito.
- Las áreas de tránsito funcionan haciendo un enlace virtual entre sus extremos, es como que de esa forma extienden el área 0.
- Los dispositivos no conocen la topología de las otras áreas.
- Si hay varios routers en una red de difusión, el que tenga un router-id mayor va a actuar como router designado. El router de backup es el que tenga el segundo mayor router-id.

Dijkstra

Es el algoritmo que usa OSPF para determinar el mejor camino.

- Primero cada router descubre su topología local.
- Despues se hace una inundación de la red con información del estado de cada router y de sus enlaces. Se llama LSA (Link State Advertising).
- Luego cada router construye un grafo de toda la red para buscar los mejores caminos.
- Finalmente cada router construye su tabla de ruteo.
- A diferencia del algoritmo de vector-distancia:
 - Se intercambian estados de enlace en lugar de tablas de ruteo.
 - Se envía la información a todos los routers en lugar de enviar a sólo los vecinos.
 - La convergencia es más rápida, hay menor carga en la red pero se necesita una capacidad de procesamiento mayor.

OSPFv3

Captura



En la foto se pueden ver los Cisco (routers azules), Mikrotik (routers azules y rojos) y el punto de captura (la lupa). En la captura se ven los paquetes OSPF que se envían entre el Mikrotik encerrado con rojo (R3) y el Cisco encerrado en azul (R1)

Rutas del Cisco:

```
IPv6 Routing Table - 15 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
C  2001:1::/64 [0/0]
   via ::, FastEthernet0/0
L  2001:1::1/128 [0/0]
   via ::, FastEthernet0/0
OE2 2001:2::/64 [110/20]
   via FE80::C602:AFF:FE0D:1, FastEthernet0/1
O  2001:3::/64 [110/11]
   via FE80::E45:23FF:FEEE:A300, FastEthernet1/0
O  2001:4::/64 [110/21]
   via FE80::E45:23FF:FEEE:A300, FastEthernet1/0
O  2001:5::/64 [110/21]
   via FE80::C602:AFF:FE0D:1, FastEthernet0/1
C  2001:A::/64 [0/0]
   via ::, FastEthernet0/1
L  2001:A::1/128 [0/0]
   via ::, FastEthernet0/1
C  2001:B::/64 [0/0]
   via ::, FastEthernet1/0
L  2001:B::1/128 [0/0]
   via ::, FastEthernet1/0
O  2001:C::/64 [110/11]
   via FE80::E45:23FF:FEEE:A300, FastEthernet1/0
   via FE80::C602:AFF:FE0D:1, FastEthernet0/1
O  2001:D::/64 [110/11]
   via FE80::E45:23FF:FEEE:A300, FastEthernet1/0
O  2001:E::/64 [110/11]
   via FE80::C602:AFF:FE0D:1, FastEthernet0/1
L  FE80::/10 [0/0]
   via ::, Null0
L  FF00::/8 [0/0]
   via ::, Null0
```

Rutas de R3:

| Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip, o - ospf, b | GATEWAY | DISTANCE |
|---|--------------------------|----------|
| # DST-ADDRESS | | |
| 0 AD0 2001:1::/64 | fe80::c601:9ff:fe68:1... | 110 |
| 1 AD0 2001:2::/64 | fe80::c602:aff:fe0d:2... | 110 |
| 2 ADC 2001:3::/64 | ether4 | 0 |
| 3 AD0 2001:4::/64 | fe80::e45:23ff:fee7:3... | 110 |
| 4 AD0 2001:5::/64 | fe80::c602:aff:fe0d:2... | 110 |

| | | | | |
|---|-----|-------------|--------------------------|-----|
| 5 | ADo | 2001:a::/64 | fe80::c601:9ff:fe68:1... | 110 |
| 6 | ADC | 2001:b::/64 | fe80::c602:aff:fe0d:2... | 0 |
| 7 | ADC | 2001:c::/64 | ether1 | 0 |
| 8 | ADC | 2001:d::/64 | ether2 | 0 |
| 9 | ADo | 2001:e::/64 | ether3 | 0 |
| | | | fe80::c602:aff:fe0d:2... | 110 |

Se puede ver que todas las rutas están bien, también comprobé que se puede llegar con ping a todos lados.

Por hacer

Intentar entender las capturas del Wireshark

IGRP

Por hacer

Agregar un poco más.

- Propietario de Cisco.
- Tiene métrica más compleja, por ancho de banda, cantidad de saltos, etc.
- Usa Bellman-Ford.

BGP

Border Gateway Protocol.

- Es el único protocolo de ruteo EGP.
- Actualmente se utiliza la versión 4.
- Usa un algoritmo de vector distancia modificado.
- EBGP por defecto tiene TTL de uno, pero IBGP está diseñado para ser ruteado por un IGP.
- Los vecinos se llaman peers.

Por hacer

- Usa UDP para algunas cosas y TCP para otras
- Cada router tiene un ID de 24 bits, por defecto es la IP de una interfaz de loopback. De lo contrario se usa una IP de alguna interfaz pero puede traer problemas si esa interfaz se cae.
- Los ID de router deben ser únicos dentro del sistema autónomo.
- Utiliza un algoritmo de vector distancia modificado.
- Para evitar bucles descarta las rutas que pasan por él mismo.
- Permite introducir restricciones o reglas políticas.
- Los sistemas autónomos pueden ser:
 - Stub: Sólo tienen una conexión.
 - De tránsito: Tienen varias conexiones.

- Multihomed: Tienen varias conexiones pero rechaza el tráfico de tránsito, esto se logra al no anunciar las redes de los demás peers.

Por hacer

Buscar atributos de rutas, como AS-Path.

Formas de aceptar rutas

1. Aceptar sólo rutas por defecto desde todos los ISP. En este caso los consumos de recursos serán muy bajos y la selección de rutas se hará utilizando el router BGP más cercano.

Hay varias formas:

- Hay que pedirle al ISP que te envíe sólamente la ruta por defecto.
- Podés filtrar las rutas que recibís y poner una ruta por defecto hacia el ISP.
- Creo que había otra más que no me acuerdo.

2. Aceptar algunas rutas más las rutas por defecto desde los ISP. En este caso el consumo de recursos de memoria y CPU será medio. El router seleccionará la ruta específica y si no la conoce lo hará a través del router BGP más cercano.

3. Aceptar todas las rutas desde todos los ISP, en este caso el consumo de recursos es alto pero en contra posición siempre se elegirá la ruta más directa.

Sincronización

Se aplica a AS de tránsito que tienen algún IGP dentro. Cuando se activa la sincronización, un AS de tránsito no anuncia por EBGP que puede alcanzar determinadas rutas hasta que se asegura de que su IGP haya convergido.

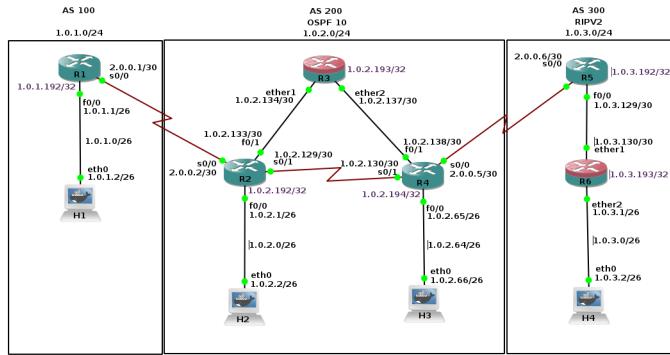
Suponiendo que hay un AS de tránsito 100 conectado a dos AS: 200 y 300. Si la sincronización no está habilitada puede pasar que por el AS de tránsito 100 anuncie al AS 300 que posee una ruta hacia el AS 100. Esto está bien siempre y cuando el IGP dentro del AS de tránsito 100 haya logrado la convergencia, de lo contrario cuando llegue un paquete desde el AS 300 hacia el AS 100 puede ocurrir que algún router dentro del AS de tránsito 100 que corre IGP pero no EBGP no conozca todavía esa ruta.

Al activar la sincronización uno se asegura que no se anuncian por EBGP rutas que temporalmente no son alcanzables. Si se deshabilita se logra la convergencia de EBGP más rápidamente a expensas de perder paquetes al principio.

Mi escenario en GNS3

Hice un escenario con un AS de tránsito conectado a dos AS de punta. El AS 200 (de tránsito) tiene OSPF y el AS 300 tiene RIP.

La idea era que los routers del AS de tránsito tengan todas las rutas para poder elegir los caminos, pero que los AS de punta reciban por BGP sólo una ruta por defecto porque no necesitan más que eso.



Rutas

R1:

```
1.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
C      1.0.1.0/26 is directly connected, FastEthernet0/0
S      1.0.1.0/24 is directly connected, Null0
C      1.0.1.192/32 is directly connected, Loopback0
      2.0.0.0/30 is subnetted, 1 subnets
C      2.0.0.0 is directly connected, Serial0/0
B*    0.0.0.0/0 [20/0] via 2.0.0.2, 00:27:25
```

R2:

```
1.0.0.0/8 is variably subnetted, 10 subnets, 4 masks
B      1.0.1.0/24 [20/0] via 2.0.0.1, 00:51:38
0 E2   1.0.3.0/24 [110/1] via 1.0.2.134, 00:10:26, FastEthernet0/1
C      1.0.2.0/26 is directly connected, FastEthernet0/0
0      1.0.2.64/26 [110/30] via 1.0.2.134, 00:10:26, FastEthernet0/1
C      1.0.2.128/30 is directly connected, Serial0/1
C      1.0.2.132/30 is directly connected, FastEthernet0/1
0      1.0.2.136/30 [110/20] via 1.0.2.134, 00:10:26, FastEthernet0/1
0      1.0.2.194/32 [110/21] via 1.0.2.134, 00:10:27, FastEthernet0/1
0      1.0.2.193/32 [110/20] via 1.0.2.134, 00:10:27, FastEthernet0/1
C      1.0.2.192/32 is directly connected, Loopback0
      2.0.0.0/30 is subnetted, 2 subnets
C      2.0.0.0 is directly connected, Serial0/0
0      2.0.0.4 [110/84] via 1.0.2.134, 00:10:31, FastEthernet0/1
S*    0.0.0.0/0 is directly connected, Null0
```

R3:

| # | DST-ADDRESS | PREF-SRC | GATEWAY | DISTANCE |
|----|------------------|-----------|-----------|----------|
| 0 | ADo 1.0.1.0/24 | | 1.0.2.133 | 110 |
| 1 | ADo 1.0.2.0/26 | | 1.0.2.133 | 110 |
| 2 | ADo 1.0.2.64/26 | | 1.0.2.138 | 110 |
| 3 | ADo 1.0.2.128/30 | | 1.0.2.138 | 110 |
| | | | 1.0.2.133 | |
| 4 | ADC 1.0.2.132/30 | 1.0.2.134 | ether1 | 0 |
| 5 | ADC 1.0.2.136/30 | 1.0.2.137 | ether2 | 0 |
| 6 | ADo 1.0.2.192/32 | | 1.0.2.133 | 110 |
| 7 | ADC 1.0.2.193/32 | 1.0.2.193 | loopback0 | 0 |
| 8 | ADo 1.0.2.194/32 | | 1.0.2.138 | 110 |
| 9 | ADo 1.0.3.0/24 | | 1.0.2.138 | 110 |
| 10 | ADo 2.0.0.0/30 | | 1.0.2.133 | 110 |
| 11 | ADo 2.0.0.4/30 | | 1.0.2.138 | 110 |

R4:

```
1.0.0.0/8 is variably subnetted, 10 subnets, 4 masks
0 E2   1.0.1.0/24 [110/1] via 1.0.2.137, 00:11:25, FastEthernet0/1
```

```

B      1.0.3.0/24 [20/0] via 2.0.0.6, 00:49:10
0      1.0.2.0/26 [110/30] via 1.0.2.137, 00:11:25, FastEthernet0/1
C      1.0.2.64/26 is directly connected, FastEthernet0/0
C      1.0.2.128/30 is directly connected, Serial0/1
0      1.0.2.132/30 [110/20] via 1.0.2.137, 00:11:25, FastEthernet0/1
C      1.0.2.136/30 is directly connected, FastEthernet0/1
C      1.0.2.194/32 is directly connected, Loopback0
0      1.0.2.193/32 [110/20] via 1.0.2.137, 00:11:26, FastEthernet0/1
0      1.0.2.192/32 [110/21] via 1.0.2.137, 00:11:26, FastEthernet0/1
      2.0.0.0/30 is subnetted, 2 subnets
0      2.0.0.0 [110/84] via 1.0.2.137, 00:11:26, FastEthernet0/1
C      2.0.0.4 is directly connected, Serial0/0
S*    0.0.0.0/0 is directly connected, Null0

```

R5:

```

1.0.0.0/8 is variably subnetted, 5 subnets, 4 masks
R      1.0.3.0/26 [120/1] via 1.0.3.130, 00:00:07, FastEthernet0/0
S      1.0.3.0/24 is directly connected, Null0
C      1.0.3.128/30 is directly connected, FastEthernet0/0
C      1.0.3.192/32 is directly connected, Loopback0
R      1.0.3.193/32 [120/1] via 1.0.3.130, 00:00:07, FastEthernet0/0
      2.0.0.0/30 is subnetted, 1 subnets
C      2.0.0.4 is directly connected, Serial0/0
B*    0.0.0.0/0 [20/0] via 2.0.0.5, 00:49:46

```

R6:

| # | DST-ADDRESS | PREF-SRC | GATEWAY | DISTANCE |
|---|------------------|-----------|-----------|----------|
| 0 | ADr 0.0.0.0/0 | | 1.0.3.129 | 120 |
| 1 | ADr 1.0.3.0/24 | | 1.0.3.129 | 120 |
| 2 | ADC 1.0.3.0/26 | 1.0.3.1 | ether2 | 0 |
| 3 | ADC 1.0.3.128/30 | 1.0.3.130 | ether1 | 0 |
| 4 | ADr 1.0.3.192/32 | | 1.0.3.129 | 120 |
| 5 | ADC 1.0.3.193/32 | 1.0.3.193 | loopback0 | 0 |
| 6 | ADr 2.0.0.4/30 | | 1.0.3.129 | 120 |

Configuraciones

Solamente pongo las de los Ciscos porque son las que más trabajo hacen.

R1:

```

interface Loopback0
 ip address 1.0.1.192 255.255.255.255
!
interface FastEthernet0/0
 ip address 1.0.1.1 255.255.255.192
 duplex auto
 speed auto
!
interface Serial0/0
 ip address 2.0.0.1 255.255.255.252
 clock rate 2000000
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 network 1.0.1.0 mask 255.255.255.0
 neighbor 2.0.0.2 remote-as 200
 no auto-summary

```

```
!
ip route 1.0.1.0 255.255.255.0 Null0 250
```

R2:

```
interface Loopback0
 ip address 1.0.2.192 255.255.255.255
!
interface FastEthernet0/0
 ip address 1.0.2.1 255.255.255.192
 duplex auto
 speed auto
!
interface Serial0/0
 ip address 2.0.0.2 255.255.255.252
 clock rate 2000000
!
interface FastEthernet0/1
 ip address 1.0.2.133 255.255.255.252
 duplex auto
 speed auto
!
interface Serial0/1
 ip address 1.0.2.129 255.255.255.252
 clock rate 2000000
!
router ospf 10
 log-adjacency-changes
 redistribute bgp 200 subnets
 passive-interface FastEthernet0/0
 passive-interface Serial0/0
 passive-interface Loopback0
 network 1.0.2.0 0.0.0.63 area 0
 network 1.0.2.128 0.0.0.3 area 0
 network 1.0.2.132 0.0.0.3 area 0
 network 1.0.2.192 0.0.0.0 area 0
 network 2.0.0.0 0.0.0.3 area 0
!
router bgp 200
 no synchronization
 bgp log-neighbor-changes
 network 1.0.2.0 mask 255.255.255.192
 network 1.0.2.128 mask 255.255.255.252
 network 1.0.2.132 mask 255.255.255.252
 network 1.0.2.192 mask 255.255.255.255
 network 2.0.0.0 mask 255.255.255.252
 neighbor 1.0.2.194 remote-as 200
 neighbor 1.0.2.194 update-source Loopback0
 neighbor 2.0.0.1 remote-as 100
 neighbor 2.0.0.1 default-originate
 neighbor 2.0.0.1 route-map DEFAULT out
 no auto-summary
!
ip route 0.0.0.0 0.0.0.0 Null0
!
ip prefix-list DEFAULT seq 5 permit 0.0.0.0/0
!
route-map DEFAULT permit 10
 match ip address prefix-list DEFAULT
```

R4:

```

interface Loopback0
 ip address 1.0.2.194 255.255.255.255
!
interface FastEthernet0/0
 ip address 1.0.2.65 255.255.255.192
 duplex auto
 speed auto
!
interface Serial0/0
 ip address 2.0.0.5 255.255.255.252
 clock rate 2000000
!
interface FastEthernet0/1
 ip address 1.0.2.138 255.255.255.252
 duplex auto
 speed auto
!
interface Serial0/1
 ip address 1.0.2.130 255.255.255.252
 clock rate 2000000
!
router ospf 10
 log-adjacency-changes
 redistribute bgp 200 subnets
 passive-interface FastEthernet0/0
 passive-interface Serial0/0
 passive-interface Loopback0
 network 1.0.2.64 0.0.0.63 area 0
 network 1.0.2.128 0.0.0.3 area 0
 network 1.0.2.136 0.0.0.3 area 0
 network 1.0.2.194 0.0.0.0 area 0
 network 2.0.0.4 0.0.0.3 area 0
!
router bgp 200
 no synchronization
 bgp log-neighbor-changes
 network 1.0.2.64 mask 255.255.255.192
 network 1.0.2.128 mask 255.255.255.252
 network 1.0.2.136 mask 255.255.255.252
 network 1.0.2.194 mask 255.255.255.255
 network 2.0.0.4 mask 255.255.255.252
 neighbor 1.0.2.192 remote-as 200
 neighbor 1.0.2.192 update-source Loopback0
 neighbor 2.0.0.6 remote-as 300
 neighbor 2.0.0.6 default-originate
 neighbor 2.0.0.6 route-map DEFAULT out
 no auto-summary
!
ip route 0.0.0.0 0.0.0.0 Null0 250
!
ip prefix-list DEFAULT seq 5 permit 0.0.0.0/0
!
route-map DEFAULT permit 10
 match ip address prefix-list DEFAULT

```

R5:

```

interface Loopback0
 ip address 1.0.3.192 255.255.255.255
!
interface FastEthernet0/0

```

```

ip address 1.0.3.129 255.255.255.252
duplex auto
speed auto
!
interface Serial0/0
ip address 2.0.0.6 255.255.255.252
clock rate 2000000
!
router rip
version 2
redistribute bgp 300 metric 1
passive-interface Serial0/0
passive-interface Loopback0
network 1.0.0.0
network 2.0.0.0
no auto-summary
!
router bgp 300
no synchronization
bgp log-neighbor-changes
network 1.0.3.0 mask 255.255.255.0
neighbor 2.0.0.5 remote-as 200
no auto-summary
!
ip route 1.0.3.0 255.255.255.0 Null0 250

```

MPLS

Multiprotocol Label Switching.

- Más rápido porque hace switching de capa 2 mirando etiquetas en lugar de ruteo de capa 3 mirando direcciones.
- Intenta conseguir las ventajas de ATM pero sin sus inconvenientes.
- No reemplaza el enrutamiento IP, aunque sea una red MPLS también es una red IP.
- Permite agrupar paquetes en flujos que recibirán todos el mismo tratamiento.
- Permite realizar ingeniería de tráfico, ya que no siempre se usa el camino más corto. También realizar QoS o Policy Routing.
- Permite realizar VPN porque realiza túneles entre dominios IP.
- La etiqueta va después de la cabecera de la capa de enlace (Ethernet, ATM, Frame-Relay, PPP) y antes de la cabecera IP. Pueden anidarse formando una pila, como si fuera una red MPLS dentro de otra.
- Al viajar por la red MPLS el campo TTL de IP no cambia, al salir se le coloca el valor adecuado.
- El enrutamiento se hace en base a la información que suministra un protocolo de ruteo, como OSPF.
- Se pueden usar switches ATM sin cambiar su hardware. Pero de todas formas es una desventaja que los routers deban ser MPLS.

Definiciones

- LSR (Label Switcher Router): Router MPLS, sobre todo los router dentro de la red MPLS, porque los de borde se llaman LER.
- LER (Label Edge Router): Router MPLS de borde.
- FEC (Forwarding Equivalence Class): Es un flujo de tráfico, está asociado a un LSP.

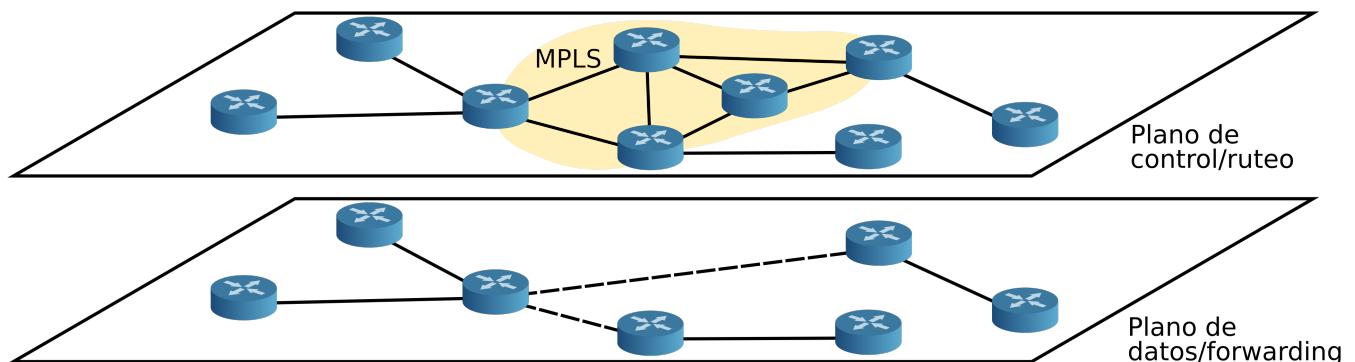
- LSP (Label Switched Path): Es un túnel virtual dentro de la red MPLS. Los túneles son unidireccionales (un solo sentido).
- LDP (Label Distribution Protocol): Protocolo que distribuye etiquetas. Permite a los LSR informar la relación FEC/etiqueta. Hay más protocolos de distribución de etiquetas, como BGP y RSVP.
- AToM (Any Transport Over MPLS): Permite usar MPLS para enviar tráfico de capa 2 como Ethernet, PPP o ATM. Por lo tanto se pueden hacer VLAN sobre MPLS.
- Upstream: Construcción de camino, se eligen las etiquetas.
- Downstream: Ya está el camino, pasan datos.

Por hacer

Terminar.

- HELLO: Protocolo de hello, empieza con UDP y pasa a TCP.
- LFIB (Label Forwarding Information Base):

Planos



- Plano de control: Donde se establece el vínculo. MPLS construye una tabla LIB de etiquetas.
- Plano de datos: Por donde se mandan datos. Están FIB y LFIB.

Funcionamiento

Cuando un paquete o flujo de paquetes llega a un LER, éste les asigna una etiqueta determinada.

Los LSR reenvían paquetes dependiendo la Label que llevan. Antes de reenviar le asignan la nueva etiqueta (swap).

En la red se generan de esta forma LSP (túneles), cada par de LSR deciden la etiqueta a usar entre ellos.

Cuando el paquete está por salir de la red MPLS, el LER extrae las etiquetas (pop).

Los LSP pueden ser creados de forma estática o por un protocolo de señalización como LDP.

Problemas a solucionar en el futuro

Estos son problemas que se intentan solucionar con MPLS pero todavía no están totalmente resueltos.

- CBR (Constraint Based Routing): Buscar caminos que satisfagan unas restricciones explícitas, como que la pérdida de paquetes sea menor a cierto valor o que el retardo sea menor a un cierto tiempo.
- Balanceo de Carga: Se intenta dividir el tráfico (de varios flujos) a través de distintos caminos.

TCP

- EL numero de secuencia empieza aleatorio, porque cuenta cada no se cuantos microsegundos, a partir de la conexión cuenta de a uno.
- Ventana deslizante pero cuenta bytes. Hasta 65000.
- Telnet usa el flag de urgente cuando se hace Ctrl-C
- En las conexiones sin estado como HTTP se suele usar RST en lugar de FIN.
- No se puede establecer una doble conexión porque se repiten los 4 cosas.
- Para hacer un firewall que impida conectarse a un host en la LAN hay que bloquear los SYN=1/ACK=0 entrantes.
- Funciona bien cuando el BDP es bajo.
- Iterativo: Atiende y corta al toque.
- Concurrente: Duplica el socket y sigue atendiendo.

Por hacer

Hacer

UDP

-

Por hacer

Hacer

GNS3

No me acuerdo bien como instalé GNS3, la cosa es que anduve y que no lo quise tocar. Pongo algunas notas de lo que hago a partir de ahora y de lo que me acuerdo que hice.

Elementos usados

- Hosts: Como host uso un Docker con la imagen de gns3-ipterm, que es un Linux bien básico con las herramientas de red básicas ya instaladas, por eso es mejor que un Alpine Linux o un Debian pelado.

Hay que instalar Docker en la computadora, después en las preferencias de GNS3 hay una sección que se llama «Docker containers», agregué el contenedor de gns3/ipterm, GNS3 se encarga de descargarlo y todo.

- Router Mikrotik: En este caso descargué un «Appliance» oficial desde la wiki de GNS3 y al mismo tiempo una imagen que se llama chr-6.41.4.img. Una vez que se hayan descargado esas dos cosas, en el GNS3 se importa el «Appliance» que te permite buscar el .img descargado y te deja todo listo para usar.

Una vez agregado, en las preferencias de GNS3 este router aparece en la parte de «QEMU VMs». Al abrir una consola el usuario es admin y no tiene contraseña, no hay que escribir nada. Para usar ipv6 en este router Mikrotik ver abajo.

- Router Cisco: Uso el modelo 3745, lo importé usando la imagen c3745-advp�servicesk9-mz.124-25d.bin que no se puede obtener gratis de forma legal. No me acuerdo como hice para agregarlo al GNS3, ahora que veo, en las preferencias de GNS3 aparece en la parte de «IOS Routers».

- Switch Cisco: Este fué dificilísimo de conseguir, no se al final que modelo tengo. Hay que conseguir:

- El Appliance cisco-iosv12.gns3a. Este es fácil de conseguir.
- Las imágenes vios_l2-adventureprisek9-m.03.2017.qcow2 y vios_l2-adventureprisek9-m.vmdk.SSA.152-4.0.55.E. Estas son muy difíciles de conseguir, el único lugar en donde las encontré es en una pagina en chino y árabe.

Hay que agregar al Appliance, al agregarlo te va a pedir que busques las dos imágenes y listo. Una vez terminado aparece en la parte de «QEMU VMs» con el nombre de Cisco IOSvL2 15.2.4055.

Otros tips

- Cuando intenté usar ipv6 en el router Mikrotik no encontraba los comandos. En mi caso pasaba que el paquete ipv6 del router no estaba habilitado, esto se puede ver al hacer system package print y viendo qué paquetes tienen una X al lado. Para habilitar un paquete hay que hacer system package enable numbers= y poniendo el número correspondiente a ese paquete (el número se ve al hacer system package print). Finalmente hay que hacer un system reboot para que se termine de activar.
- Cuando te pasan un proyecto con Ciscos adentro puede que ande lento, para acelerarlo hay que hacer click derecho en un Cisco cualquiera y poner Auto Idle-PC value.

Wireshark

Ejemplos de filtros

ARP e ICMP que salgan o lleguen a una MAC:

```
(arp || icmp) && (eth.src == 08:9e:01:43:f2:eb || eth.dst == 08:9e:01:43:f2:eb)
```

Tramas Ethernet de broadcast:

```
eth.dst == ff:ff:ff:ff:ff:ff
```

Paquetes IP de broadcast:

```
ip.dst == 192.168.0.255
```

ARP de respuesta:

```
arp.opcode == reply
```

ICMP de respuesta:

```
icmp.type == 0
```

ICMP de request:

```
icmp.type == 8
```

Comandos

Linux

Básico

Nota

Está la guía de comandos para Linux.

ARP

Para hacer ARP gratuitos:

```
arping -A -c 100000 -I enp4s0 {mi IP}
```

NAT

```
sudo bash -c "echo 1 > /proc/sys/net/ipv4/ip_forward"  
modprobe iptable_nat
```

Ver tabla:

```
iptables -t nat -n -L
```

Agregar regla con IP de salida:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to {IP salida}
```

Servidor DHCP

Instalar:

```
sudo apt intall isc-dhcp-server
```

Editar /etc/default/isc-dhcp-server y configurar interfaz a usar:

```
INTERFACESv4="eth0"
```

Editar /etc/dhcp/dhcpd.conf:

```
default-lease-time 600; # 10 minutes  
max-lease-time 7200; # 2 hours  
  
ddns-update-style none;  
authoritative;  
  
option domain-name-servers 8.8.8.8;  
option broadcast-address 192.168.1.255;  
option routers 192.168.1.1;  
option subnet-mask 255.255.255.0;  
  
subnet 192.168.1.0 netmask 255.255.255.0  
{  
range 192.168.1.100 192.168.1.200;  
}
```

Iniciar:

Comandos

```
sudo /etc/init.d/isc-dhcp-server restart
```

Si hay problemas usar:

```
less /var/log/syslog  
sudo systemctl status isc-dhcp-server.service
```

Cliente DHCP

Desactivar network-manager, sacarse todas las IP y pedir por DHCP:

```
sudo service network-manager stop  
sudo ip flush dev eth0  
sudo dhclient eth0
```

Servidor PPPoE

NAT:

```
echo 1 > /proc/sys/net/ipv4/ip_forward  
iptables -t nat -A POSTROUTING -o x -j SNAT --to x.x.x.x  
iptables -t nat -n -L
```

Instalar rp-pppoe:

```
tar -xvzf rp-pppoe-x.tar.gz  
cd rp-pppoe-x/src  
.configure  
make  
make install
```

Configurar /etc/ppp/pppoe-server-options:

```
requirechap  
lcp-echo-interval 30 # cada cuanto pregunta si está la conexión activa  
lcp-echo-failure 4 # intentos  
netmask 255.255.255.255  
ms-dns {IP DNS}
```

Configurar /etc/ppp/chap-secrets:

```
# client server secret IP addresses  
"usuario1" * "contraseña1" {IP fija que queres dar}  
"usuario2" * "contraseña2" *
```

Configurar permisos:

```
chmod 600 /etc/ppp/chap-secrets
```

Configurar direcciones dinámicas en nano /etc/ppp/direcciones, una IP por línea o un rango como 192.168.50.2-30.

Iniciar servidor, dando interfaz e IP de la interfaz por donde sale el PPPoE. También dar MTU a usar:

```
pppoe-server -C {nombre_del_servidor} -L {IP interfaz} -p /etc/ppp/direcciones  
-I {interfaz} -m 1412
```

Cliente PPPoE

Seguir los pasos que te pide pppoeconf.

IPv6

Automáticamente toma una dirección link-local y toma una dirección global si recibe un RA.

Para hacer ping a una link-local hay que dar la interfaz:

```
ping6 {IPv6}  
ping6 {IPv6 link-local} -I {interfaz}
```

Por hacer

Ver si eso de dar la interfaz pasa por algo de linux o porque ip6 funciona así

Cisco

- Se usa puerto de consola serial a 9600 baudios
- Para ayuda ir poniendo ?.
- Para ir volviendo atrás poner exit.

Básico

Pasar al modo privilegiado:

```
router> enable
```

Mostrar cosas:

```
router# show ip interface brief  
router# show ipv6 interface brief  
router# show ip interface f0/0  
router# show ipv6 interface f0/0  
router# show ip route  
router# show running-config
```

Configurar IP:

```
router# configure terminal  
router(config)# interface FastEthernet0/0  
router(config-if)# ip address 192.168.1.1 255.255.255.0
```

Configurar IP secundaria:

```
router(config-if)# ip address 192.168.2.1 255.255.255.0 secondary
```

Habilitar interfaz:

```
router(config-if)# no shutdown  
router(config-if)# exit
```

Configurar rutas:

```
router(config)# ip route 0.0.0.0 0.0.0.0 {IP gateway}  
router(config)# ip route {IP destino} {máscara destino} {IP gateway}  
  
router(config)# ipv6 route ::/0 {IP gateway}  
router(config)# ipv6 route {IP destino}/{máscara destino} {IP gateway}
```

Crear loopback y dar IP:

```
(config)# interface loopback 0  
(config-if)# ip address 7.0.0.4 255.0.0.0
```

Para guardar la configuración de inicio:

```
router# copy running-config startup-config
```

Para borrar la configuración de inicio encontré tres formas, en GNS3 no anda ninguna:

Comandos

```
router# erase startup-config
router# erase nvram:
router# write erase
```

Para reiniciar en GNS3 usar el botón de «Stop», en un dispositivo físico:

```
router# reload
```

Para reiniciar la configuración en GNS3, agregar un nuevo router, click derecho, exportar su configuración, hacer click derecho en el router a reiniciar, imporar configuración.

NAT

Configurar cual es la interna y cual externa:

```
router# configure terminal
router(config)# interface FastEthernet0/0
router(config-if)# ip nat inside
router(config-if)# exit
router(config)# interface FastEthernet0/1
router(config-if)# ip nat outside
router(config-if)# exit
```

Configurar lista de acceso con un numero mayor a 100 protocolo IP, para cualquier origen y cualquier destino:

```
router(config)# access-list 101 permit ip any any
```

Configurar pool de rango de IPs a usar para nateo (las de afuera, normalmente una sola):

```
router(config)# ip nat pool {nombre} {IP inicial} {IP final} netmask {mascara}
```

Terminar:

```
router(config)# ip nat inside source list 101 pool {nombre pool} overload
```

Mostrar info:

```
router(config)# exit
router# show ip nat translations
```

Servidor DHCP

Primero configurar IPs y rutas.

Dar de alta dhcp, con «no» se da de baja:

```
router# configure terminal
router(config)# service dhcp
```

Crear pool de IPs, excluir IPs de la red a no asignar y dar red a utilizar:

```
router(config)# ip dhcp excluded-address {IP minima} {IP maxima}
router(config)# ip dhcp pool {nombre}
```

Configurar parametros, la red es el rango de IPs a dar:

```
router(dhcp-config)# network {Numero red} {mascara}
router(dhcp-config)# domain-name {IP DNS}
router(dhcp-config)# default-router {IP gateway}
router(dhcp-config)# lease {dias}
```

Configurar tiempo de comprobacion de si la IP esta asignada:

```
router(dhcp-config)# exit
router(config)# ip dhcp timeout {milisegundos}
```

Comprobacion de configuracion:

Comandos

```
router(config)# exit  
router# exit  
  
router# show ip dhcp binding  
router# show ip dhcp conflict  
router# show ip dhcp server statistics
```

Mostrar debug:

```
router# debug ip dhcp server events|packet|linkage
```

Enlace serial

- DCE: (Hembra), Configura velocidad.
- DTE: (Macho), Recibe velocidad.

Para ver interfaces seriales en Cisco (muestra si es DCE o DTE):

```
router# show controllers Serial0/0/0
```

Para configurar DCE:

```
router# configure terminal  
router(config)# interface serial 0/0/0  
router(config-if)# clock rate 125000  
router(config-if)# ip address {ip}
```

Para configurar DTE en Cisco no hay nada especial. Directamente te pones IP.

Servidor PPPoE

Configurar salida al exterior:

```
router# configure terminal  
router(config)# interface fastEthernet 0/0  
router(config-if)# ip address {IP salida}  
router(config-if)# no shutdown  
router(config-if)# exit  
router(config)# ip route 0.0.0.0 0.0.0.0 {gateway}
```

PPPoE:

```
router# configure terminal  
router(config)# username {usuario cliente} password {pass cliente}  
router(config)# bba-group pppoe global  
router(config)# virtual-template 20
```

Configurar interfaz:

```
router(config)# interface fastEthernet 0/1  
router(config-if)# pppoe enable
```

Configurar PPPoE:

```
router(config)# interface virtual-template 20  
router(config-if)# mtu 1492  
router(config-if)# ppp mtu adaptive  
router(config-if)# ip unnumbered fastEthernet 0/1  
router(config-if)# peer default ip address pool pppoepool  
router(config-if)# ppp authentication {no me acuerdo si era "pap" o "chap"}  
router(config-if)# ppp chap hostname {nombre servidor}  
router(config-if)# ppp chap password 0 {pass servidor}  
router(config-if)# exit  
router(config)# ip local pool pppoepool 10.10.10.1 10.10.10.200
```

Comandos

Falta hacer NAT para que pueda salir afuera.

VLAN (switch)

Mostrar VLANs activas:

```
switch# show vlan
```

Agregar VLAN:

```
switch(config)# vlan 2
switch(config-vlan)# name {nombre_vlan}
switch(config-vlan)# exit
```

Agregar interfaces a una VLAN:

```
switch(config)# interface range fastEthernet 0/{x}-{y}
switch(config-if)# switchport mode access
switch(config-if)# switchport access vlan 2
```

Agregar trunk:

```
switch(config)# interface fastEthernet 0/{x}
switch(config-if)# switchport switchport trunk encapsulation dot1q
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk native vlan 1
switch(config-if)# switchport trunk allowed vlan all
switch(config-if)# switchport trunk allowed vlan add 2
switch(config-if)# switchport trunk allowed vlan remove 3
```

VLAN (router)

Rutear con Cisco en trunk:

```
router(config)# interface FastEthernet0/0
router(config-if)# sub-interface FastEthernet0/0.2
router(config-if)# encapsulation dot1q 2
router(config-if)# ip address 192.168.1.1 255.255.255.0
router(config-if)# sub-interface FastEthernet0/0.3
router(config-if)# encapsulation dot1q 3
router(config-if)# ip address 192.168.2.1 255.255.255.0
```

IPv6

Habilitar ruteo IPv6:

```
router(config)# ipv6 unicast-routing
```

Habilitar dirección link-local bajo una interfaz:

```
router(config)# interface {interfaz}
router(config-if)# ipv6 enable
```

Habilitar RA en una interfaz y configurar el intervalo en segundos:

```
router(config)# interface {interfaz}
router(config-if)# no ipv6 nd suppress-ra
router(config-if)# ipv6 nd ra-interval {intervalo}
```

Me pasó a veces que RA viene activado en las interfaces por defecto, es posible que si uno no quiere RA, haya que deshabilitarlo manualmente:

```
router(config)# interface {interfaz}
router(config-if)# ipv6 nd suppress-ra
```

Comandos

En routers nuevos puede que los comandos sean:

```
router(config)# interface {interfaz}
router(config-if)# no ipv6 nd ra suppress
router(config-if)# ipv6 nd ra interval {intervalo}
```

6to4



Activar ruteo IPv6:

```
router(config)# ipv6 unicast-routing
```

Dar IPv4 a la interfaz WAN:

```
router(config)# interface FastEthernet {if_wan}
router(config-if)# ip address {ipv4_if_wan} {ipv4_mask_if_wan}
router(config-if)# no shutdown
router(config-if)# exit
```

Crear tunel y ponerle una IPv6 que está en la red 2002:{ipv4}:0000::/64, generalmente la ::1 e indicar que sale por la {if_wan}:

```
router(config)# interface Tunnel2002
router(config-if)# no ip address
router(config-if)# no ip redirects
router(config-if)# no shutdown
router(config-if)# ipv6 address {ipv6_if_wan}/128
router(config-if)# tunnel source FastEthernet {if_wan}
router(config-if)# tunnel mode ipv6ip 6to4
router(config-if)# exit
```

Configurar la interfaz con IPv6 que da a la LAN, tiene que estar en una subred, generalmente también es la ::1, en tal caso es 2002:{ipv4}:{subred}:1/64. También habilitar RA:

```
router(config)# interface FastEthernet {if_lan}
router(config-if)# ipv6 address {ipv6_if_lan} {ipv6_mask_if_lan}
router(config-if)# no shutdown
router(config-if)# no ipv6 nd suppress-ra
```

Configurar las rutas, la primera es una ruta IPv4 común para que se llegue a destino por IPv4. La segunda no se bien por qué está. La tercera es el gateway IPv6, en el caso en el que hayan varios túneles hacia varias redes IPv6 habría que separar las rutas en vez de tener una que vaya a ::/0. Pero en el caso más simple es:

```
router(config)# ip route 0.0.0.0 0.0.0.0 {ipv4_if_gw}
router(config)# ipv6 route 2002::/16 Tunnel2002
router(config)# ipv6 route ::/0 {ipv6_if_wan_destino}
```

RIP

RIPv2

Hay que decirle que version usar.

```
router(config)#router rip
router(config-router)#version 2
```

Comandos

Desactivar la summarización de redes, por lo tanto al usar el comando network se publican las subredes que pertenezcan a esa red en lugar de la red completa como una sola:

```
router(config-router)#no auto-summary
```

Agregar redes en donde trabajar. De paso publica esas redes, como no pide máscara se fija en la clase de la red para determinar la máscara, summarizando las redes:

```
router(config-router)#network {red}
```

Pasivar una interfaz:

```
R6(config-router)#passive-interface {interfaz}
```

Distribuir más redes, usar metricas menores a 15:

```
router(config-router)#redistribute static  
router(config-router)#redistribute static metric {metrica}
```

Redistribuir EBGP si es necesario, usar metricas menores a 15:

```
router(config)# router rip  
router(config-route)# redistribute bgp {numero_as} metric {metrica}
```

Redistribuir IBGP si es necesario, usar metricas menores a 15:

```
router(config)# router rip  
router(config-route)# redistribute bgp {numero_as} metric {metrica}  
router(config-route)# exit  
router(config)# router bgp {numero_as}  
router(config-route)# bgp redistribute-internal
```

RIPng

Primero dar IPv6 estáticas a cada interfaz y borrar las rutas si es que hay.

Después habilitar RIP en las interfaces que uno quiera, asignándole un nombre cualquiera al proceso:

```
router(config)# interface {interfaz}  
router(config-if)# ipv6 rip {nombre_proceso} enable  
router(config-if)# exit  
router(config)# ipv6 router rip {nombre_proceso}  
router(config-if)# redistribute connected
```

OSPF

OSPFv2

Crear loopback:

```
router(config)# interface loopback 0  
router(config-if)# ip address {ip_loopback} {mascara}
```

Crear un proceso OSPF, por ejemplo poner el número 100:

```
router(config)# router ospf {id_proceso}
```

Agregar redes en donde trabajar. De paso publica esas redes, con la máscara dada. La mascara debe ser invertida, por ejemplo 255.255.255.240 se vuelve 0.0.0.15, si se suman queda 255 en cada octeto:

```
router(config-route)# network {red} {mascara_invertida} area 1
```

Mascara invertida: 0.255.255.255 publica la 10.X.X.X.

Pasivar interfaces:

```
router(config-route)# passive-interface {interfaz}  
router(config-route)# passive-interface loopback0
```

Comandos

Redistribuir EBGP si es necesario:

```
router(config)# router ospf {proceso}
router(config-route)# redistribute bgp {numero_as} subnets
```

Redistribuir IBGP si es necesario:

```
router(config)# router ospf {proceso}
router(config-route)# redistribute bgp {numero_as} subnets
router(config-route)# exit
router(config)# router bgp {numero_as}
router(config-route)# bgp redistribute-internal
```

OSPFv3

Primero se debe asignar una IPv4 al loopback, esta IP es usada por OSPF para identificar a este router:

```
router(config)# interface loopback 0
router(config-if)# ip address {ipv4_loopback} {mascara}
router(config-if)# exit
```

Después se debe entrar a una interfaz de WAN y activar OSPF, se le debe asignar un ID de proceso cualquiera mientras que sea un número, yo uso 1:

```
router(config)# interface {interfaz}
router(config-if)# ipv6 ospf {id_proceso} area 0
```

Para indicar qué rutas se deben publicar:

```
router(config-if)# ipv6 router ospf {id_proceso}
router(config-rtr)# redistribute static
router(config-rtr)# redistribute connected
router(config-rtr)# exit
```

Si se quieren agregar más interfaces WAN se puede compartir el ID del proceso, no es necesario indicar las rutas a distribuir de nuevo:

```
router(config)# interface FastEthernet {interfaz}
router(config-if)# ipv6 ospf {id_proceso} area 0
```

BGP

Solamente hay un proceso, hay que darle el numero de AS:

```
router(config)# router bgp {numero_as}
```

Dar redes a anunciar, a diferencia de otros protocolos esto no indica las interfaces que corren el protocolo, solamente las redes a anunciar:

```
router(config-router)# network {red} mask {mascara}
```

Agregar vecino peer:

```
router(config-router)# neighbor {ip_destino} remote-as {as_destino}
```

Agregar peer estableciendo dirección de loopback de destino por si una de sus interfaces se cae, normalmente se hace en IBGP:

```
router(config-router)# neighbor {ip_loopback_destino} remote-as {as_destino}
```

Originar tráfico hacia peer desde mi propio loopback, en caso en que este router tenga varias interfaces de salida hacia el peer. Si se cae una de las interfaces, la comunicación comienza a originarse desde otra interfaz, el destino verá que la IP de origen cambió y rechazará la conexión:

```
router(config-router)# neighbor {ip_loopback_destino} update-source loopback0
```

Comandos

Publicar una ruta sumarizada que engloba a todas las redes dentro del AS, no se pueden publicar rutas que no están en la tabla de ruteo el null0 asegura que si llega un paquete y no hay una ruta más específica el paquete se descarta.:

```
router(config)# ip route {red} {mascara} null0 250
```

```
router(config)# router bgp {numero_as}
```

```
router(config)# network {red} mask {mascara}
```

Configurar para que un peer reciba solamente la ruta por defecto, esto se podría en un AS de tránsito para darle servicio a un AS que no es de tránsito:

```
router(config)# ip route 0.0.0.0 0.0.0.0 null0 250
```

```
router(config)# ip prefix-list DEFAULT permit 0.0.0.0/0
```

```
router(config)# route-map DEFAULT permit 10
```

```
router(config-route-map)# match ip address prefix-list DEFAULT
```

```
router(config-route-map)# exit
```

```
router(config)# router bgp {numero_as}
```

```
router(config-router)# neighbor {ip_destino} route-map DEFAULT out
```

```
router(config-router)# neighbor {ip_destino} default originate
```

Nota

Supongo que ese último bloque de comandos se puede resumir, el último comando propaga la ruta por defecto a un peer en particular, el problema es que cuando lo probé el peer recibía todas las rutas BGP más la por defecto.

Todos los comandos anteriores lo que hacen es algo así como un filtro de rutas para el peer, al cual se le daría solamente la ruta por defecto. El problema que tuve es que cuando lo probé no propagaba nada, entonces agregué el último comando que propaga la ruta por defecto y quedó como me gustaría.

Ver estado de sesiones BGP:

```
router# show ip bgp summary
```

Ver tabla de enrutamiento BGP:

```
router# show ip bgp
```

Ver vecinos y tipo y cantidad de mensajes intercambiados:

```
router# show ip bgp neighbors
```

MPLS

No hacemos la implementación, pero pongo algunos comandos que dan información.

Ver las interfaces sobre las que funciona MPLS-LDP:

```
show mpls interfaces
```

Muestra los parámetros que está usando el protocolo en este equipo:

```
show mpls ldp parameters
```

Mostrar los vecinos MPLS:

```
show mpls ldp neighbor
```

Mostrar la tabla de etiquetas:

```
show mpls ldp binding
```

Comandos

Mostrar la tabla de forwarding:

```
show mpls forwarding-table
```

No sé que hace este:

```
show mpls ip binding
```

Mikrotik

- Generalmente usuario admin sin contraseña.
- Se usa puerto de consola serial a 115200 baudios
 - Modelo 230: 9600 baudios
- Para ayuda ir poniendo ?.
- Si se pone un comando incompleto se entra a un sub-menu, para ir volviendo atrás poner ...

Mostrar cosas:

```
ip address print  
ip route print  
  
ipv6 address print  
ipv6 route print  
  
interface print
```

Habilitar o deshabilitar interfaz:

```
interface print  
interface enable numbers={numero_interfaz}  
interface disable numbers={numero_interfaz}
```

Agregar IP:

```
ip address add interface=eth0 address=192.168.1.1/24  
ip address remove numbers=1  
  
ipv6 address add interface=eth0 address=2001:AA::2/64 advertise=no  
ipv6 address remove numbers=1
```

El advertise al agregar una IPv6 indica si se habilita RA en esa interfaz. El intervalo por defecto es bastante lento y puede parecer que el RA no anda.

Agregar rutas:

```
ip route add dst-address=0.0.0.0/0 gateway=192.168.1.1  
ip route remove numbers=1  
  
ipv6 route add dst-address=::/0 gateway=2001:A::1/64  
ipv6 route remove numbers=1
```

Configurar DNS:

```
ip dns set servers=8.8.8.8
```

Crear loopback y dar IP:

```
interface bridge add name=loopback0  
ip address add interface=loopback0 address=1.1.1.1
```

Ver y habilitar paquetes:

```
system package print  
system package enable numbers={numero_paquete}  
system reboot
```

Comandos

Reiniciar configuración:

```
system reset-configuration
```

NAT

```
ip firewall nat add chain=srcnat action=masquerade out-interface={interfaz salida}
```

Servidor DHCP

```
ip pool add name=dhcppool ranges=10.0.0.10-10.0.0.50
```

```
ip dhcp-server network add address=10.0.0.0/8 gateway=10.0.0.1 dns-server=8.8.8.8
```

```
ip dhcp-server add name=dhcpserver interface=ether2 address-pool=dhcpool disabled=no
```

Servidor PPPoE

Poner NAT:

```
ip firewall nat add chain=srcnat action=masquerade out-interface={interfaz salida}
```

Configurar rango:

```
ip pool add name={nombre pool} ranges={primera IP} {ultima IP}
```

Agregar perfil cliente:

```
ppp profile add name={nombre perfil} local-address {IP placa interna}  
remote-address={nombre pool} dns-server={IP DNS} rate-limit=1024kbps
```

Cargar clientes:

```
ppp secret add name={nombre usuario} password={pass usuario} service=pppoe  
profile={nombre perfil}
```

Asignar servicio a interfaz:

```
interface pppoe-server server add service-name={nombre servidor}  
interface={interfaz interna} default-profile={nombre perfil} disabled=no
```

Ver problemas:

```
log print
```

IPv6

Puede que el paquete ipv6 esté deshabilitado, ver arriba cómo se hace para habilitarlo.

Para cambiar el intervalo, imprimir la lista de RA activos y configurar el intervalo. El intervalo tiene un valor mínimo y uno máximo:

```
ipv6 nd print  
ipv6 nd set ra-interval=10s-20s numbers=0
```

6to4



Comandos

Primero agregar la IPv4 de WAN y agregar ruta por defecto para el gateway ipv4:

```
ip address add interface={if_wan} address={ipv4_if_wan}/{ipv4_mask_if_wan}
ip route add dst-address=0.0.0.0/0 gateway={ipv4_if_gw}
```

Configurar la interface 6to4:

```
interface 6to4 add mtu=1280 name={nombre_tunel} local-address={ipv4_if_wan} remote-address=
```

Asociar la ipv6 calculada previamente a la interfaz creada:

```
ipv6 address add address={ipv6_if_wan}/3 interface={nombre_tunel}
```

Agregar una ruta ipv6 por defecto a través del túnel:

```
ipv6 route add dst-address=2000::/3 gateway=::{ipv4_if_wan_destino}%{nombre_tunel}
```

Configurar la interfaz con IPv6 que da a la LAN, a diferencia de Cisco no tiene que estar en una subred, la {ipv6_if_lan} puede ser igual a {ipv6_if_wan}. Pero es mejor ponerlo en una subred:

```
ipv6 address add address={ipv6_if_lan}/64 interface={if_lan} advertise=yes disabled=no
```

RIP

RIPv2

Por defecto es V2. Nunca probé usar V1.

Agregar redes en donde trabajar y redes a publicar:

```
routing rip network add network={red}/{mascara}
```

Pasivar interfaces:

```
routing rip interface add passive=yes interface={interfaz}
```

RIPng

Agregar redes en donde trabajar y redes a publicar:

```
routing ripng interface add interface={interfaz}
```

Cambiar tiempo de publicación en segundos (agregar s al final):

```
routing ripng set update-timer={tiempo}s
```

Hay más opciones a configurar, se muestran con:

```
routing ripng print
```

Para distribuir otros rutas:

```
routing ripng set distribute-default={never|always|if-installed}
routing ripng set redistribute-static={yes|no}
routing ripng set redistribute-bgp={yes|no}
routing ripng set redistribute-ospf={yes|no}
```

OSPF

OSPFv2

Crear loopback y dar IP:

```
interface bridge add name=loopback0
ip address add interface=loopback0 address={ip}/{mascara}
```

Crear instancia OSPF (no sé como se relaciona el nombre con los números usados en Cisco):

```
routing ospf instance add name=default
```

Comandos

Agregar redes en donde trabajar y redes a publicar:

```
routing ospf network add network={red}/{mascara} area={area}
```

Pasivar interfaces:

```
routing ospf interface add interface={interfaz} passive=yes  
routing ospf interface add interface=loopback0 passive=yes
```

OSPFv3

Configurar OSPFv3, no sé por qué los comandos son tan raros. Se debe dar una IPv4 cualquiera como ID para este router:

```
routing ospf-v3 instance set default redistribute-connected=as-type-1 router-id={ipv4_id}  
routing ospf-v3 area set {area} instance=default
```

Agregar redes en donde trabajar:

```
routing ospf-v3 interface add interface={interfaz} area={area} network-type=broadcast
```

Agregar redes pasivas:

```
routing ospf-v3 interface add interface={interfaz} area={area} network-type=broadcast passive
```

Links

- <https://wiki.wireshark.org/FrontPage>