CIDR (Classless InterDomain Routing)

* Creado por el IETF
* Este esquema de direccionamiento brinda las mismas ventajas que VLSM (Máscaras flexibles, subnetting, menos desperdicio del espacio de direccionamiento) y además permite generar superredes (supernetting)

**Ejemplo:** *Tengo una granja de 1,000 servidores y quiero agruparlos en una sola red.*

*>>Solución*

Descartamos redes A y B en otras palabras

A -> X

B -> X

C -> Son las que utilizaremos

|  |  |
| --- | --- |
| Id red | Id host |
| 3 bytes | 1 byte |

El número de IPs que podemos utilizar son 28 – 2 = 254 IPs

Si lo dividimos en 4 bloques clase C entonces

|  |
| --- |
| 200.1.12.0/24 |
| 200.1.13.0/24 |
| 200.1.14.0/24 |
| 200.1.13.0/24 |

**Reglas para crear una super red**

1. La cantidad de bloques utilizados debe ser múltiplo de una potencia de 2
2. Los bloques deben ser adyacentes (sin huecos entre ellos)
3. El primer bloque que represente a la superred deberá un 3we byte divisible entre el número de bloques

A que nos referimos con que sean adyacentes por ejemplo

|  |
| --- |
| 199.1.3.0/24 tiene un rango de 199.1.3.0 – 199.1.3.255 |
| 199.1.5.0/24 |
| 199.1.6.0/30 tiene un rango de 199.1.6.0 – 199.1.6.3 |
| 199.1.7.0/24 |

Podemos observar que no tenemos 199.1.4.0/24 además que el segmento 199.1.6.0/**30** solo tenemos dos bytes para host

Retomando a nuestro ejemplo anterior nuestra super red será la 200.1.12.0/24 además todas en el primer byte son iguales por tanto tienen 8 bytes iguales, el segundo byte todos son iguales portanto hay 8 bytes idénticos o sea

8 + 8

O bien lo representamos de la siguiente manera

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 12 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 13 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 14 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 15 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

Podemos observar que de 128 a 4 los bytes son iguales, entonces difieren 2 y 1 por tanto hasta sexto byte se enciende entonces tenemos 8 + 8 + 6 = 22

Calculamos el rango útil que lo obtenemos de 200.1.12.0/24 y termina en 200.1.15.0/24

200.1.12.1 hasta 200.1.15.254 retomando la fórmula 28-2 = 254

El broad cast es 200.1.15.255

Finalmente calculamos la máscara que la obtenemos de 8 + 8 + 6 = 22

11111111.11111111.11111100.0000000

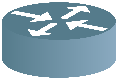
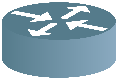
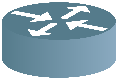
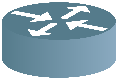
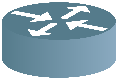
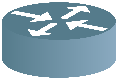
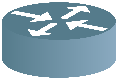
Convertimos a decimal y obtenemos

255.255.252.0 -> máscara

**Enrutamiento:**

Enrutamiento es el proceso buscar un camino entre todos los posibles con menor métrica posible.

* Sistema autónomo (A.S.): Se define como un conjunto de redes administrado por una sola entidad e independiente con su propia política de enrutamiento. Se identifica mediante un Número de sistema autónomo, el cual es registrada por la IANA
* Tipos de enrutamiento:
  + Interior: Se da dentro de un A.S.
    - Estático
    - Dinámico
  + Exterior: Se da entre AS’s
    - Estático
    - Dinámico



AS1 AS2

Sea enrutamientos interiores o exteriores todo se basa en algoritmos:

**Algoritmos usados por los protocolos de enrutamiento**

* Algoritmo de Dijkstra: Basado en el estado del enlace (costo), usado por el protocolo OSPF

Para el cálculo de los costos tenemos una fórmula

Nota: C es la capacidad del canal

* Algoritmo de Bellman – Ford: Basado en el número de saltos para llegar a un destino, usado por el protocolo RIP

**Algoritmo de Dijkstra**

* Sea ‘S’ el nodo raíz para el cual se calculará la tabla con los costos mínimos de todo el grafo
* ‘M’ Conjunto de notos incorporados por el algoritmo (1 en )
* dij costo asociado al enlace entre los nodos *i* y *j*
* Dn es el costo asociado a la trayectoria desde el nodo S al nodo ‘n’

Dn = min[dn, dw + dnw]

* W = ultimo nodo incorporado a ‘M’

**EJEMPLO:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M | D1 | Trayectoria | D2 | Trayectoria | D3 | Trayectoria | D4 | Trayectoria | D6 | Trayectoria |
| 5 |  | - |  | - | 1 | 5-3 | 1 | 5-4 | 2 | 5-6 |
| 5,3 | 2 | 5-3-1 | 3 | 5-3-2 |  |  | 1 | 5-4 | 2 | 5-6 |
| 5,3,4 | 2 | 5-3-1 | 3 | 5-3-2 |  |  |  |  | 2 | 5-6 |
| 5,3,4,1 |  |  | 3 | 5-3-2 |  |  |  |  | 2 | 5-6 |
| 5,3,4,1,6 |  |  | 3 | 5-3-2 |  |  |  |  |  |  |
| 5,3,4,1,6,2 |  |  |  |  |  |  |  |  |  |  |

**Tabla de enrutamiento del nodo 5**

|  |  |  |
| --- | --- | --- |
| **Destino** | **Métrica** | **Siguiente salto** |
| 1 | **2** | **3** |
| **2** | **3** | **3** |
| **3** | **1** | **3** |
| **4** | **1** | **4** |
| **5** | **0** | **-** |
| **6** | **2** | **6** |

**Ejercicio calcular para el resto del nodo**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M | D2 | Trayectoria | D3 | Trayectoria | D4 | Trayectoria | D5 | Trayectoria | D6 | Trayectoria |
| 1 | 2 | 1-2 |  | 1-3 | 5 | 1-4 |  | - |  | - |
| 1,3 | **2** | 1-2 |  |  | 4 | 1-3-4 | 2 | 1-3-5 |  | - |
| 1,3,5 |  |  |  |  | 3 | 1-3-5-4 | **2** | 1-3-5 | 6 | 1-3-5-6 |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

**Bellman – Ford**

* Toma como métrica la cantidad de saltos usados para llegar a un nodo destino
* Sea ‘S’ el nodo raíz para el que se obtendrá la tabla con los costos mínimos
* H=El número de saltos usados por el algoritmo
* = costo asociado al enlace entre el nodo i y el nodo j

=Costo asociado de ir desde el nodo “S” al nodo “n” con “h” saltos

Tenemos el siguiente grafo

2

8

1

1

2

3

3

5

1

2

Tomando como referencia al nodo 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | + |  | = min | ∞ + 0 |
|  | + |  | ∞ + 2 |
|  | + |  | ∞ + 1 |
|  | + |  | 0 + 5 |
|  | + |  | ∞ + ∞ |
|  | + |  | ∞ + ∞ |

= 5

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| H | D1 | Trayectoria | D2 | Trayectoria | D3 | Trayectoria | D4 | Trayectoria | D5 | Trayectoria | D6 | Trayectoria |
| 0 | ∞ | - | ∞ | - | ∞ | - | 0 | 4 | ∞ | - | ∞ | - |
| 1 | 5 | 4-1 | 3 | 4-2 | 3 | 4-3 | 0 | 4 | 1 | 4-5 | 8 | 4-6 |
| 2 | 4 | 4-3-1 | 3 | 4-2 | 3 | 4-3 | 0 | 4 | 1 | 4-5 | 3 | 4-5-6 |
| 3 | 3 | 4-5-3-1 | 3 | 4-2 | 2 | 4-5-3 | 0 | 4 | 1 | 4-5 | 3 | 4-5-6 |
| 4 | 3 | 4-*5*-3-1 | 3 | 4-*2* | 2 | 4-*5*-3 | 0 | 4 | 1 | 4-*5* | 3 | 4-*5*-6 |

|  |  |  |
| --- | --- | --- |
| Destino | Métrica | Siguiente salto |
| 1 | 3 | 5 |
| 2 | 3 | 2 |
| 3 | 2 | 5 |
| 4 | 0 | - |
| 5 | 1 | 5 |
| 6 | 3 | 5 |

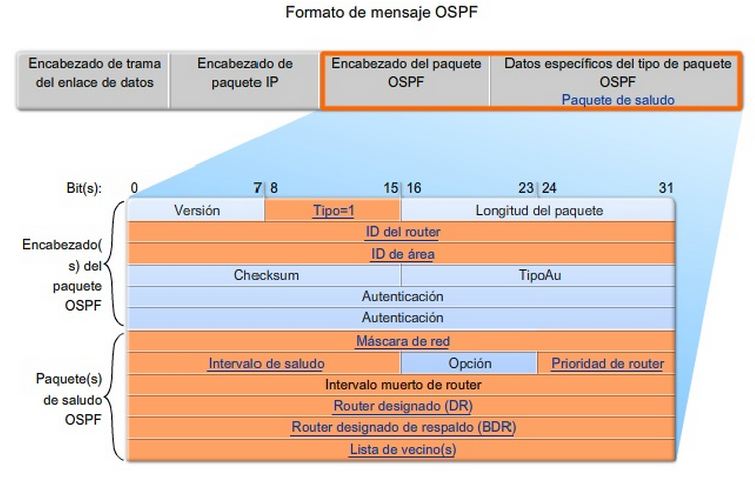
**Protocolo RIP(Routing Information Protocol)**

* En un protocol de enrutamiento interior (IGP)
* Utiliza como métrica el número de saltos utilizado para llegar a los segmentos de las redes vecinas (algoritmo de Bellman - Ford)
* No puede ser implantado en redes muy grandes (más de 15 saltos de diámetro) pues no es capaz de detectar rutas circulares
* Los enrutadores se comunican mediante datagramas (protocolo UDP) enviados al puerto 520 a la dirección de broadcast (255.255.255.255)
* Un enrutador RIP enviará periódicamente (c/30 seg) publicando su tabla de enrutamiento completa
* Los enrutadores que reciben mensajes RIP actualizarán sus tablas de enrutamiento cuando descubran nuevas rutas con menos métrica

Nacido cerca de los 80s por Xerox

**OSPF (Open Shortest Path First)**

* Es un protocolo de enrutamiento interior (IGP) que basa su funcionamiento en el algoritmo de Dijkstra (estado-enlace)
* Puede usar los 3 tipos de direccionamiento unicast (CLASES, VLSM, CIDR) y soporta sumarizacion de rutas
* OSPF puede ser implantado en redes grandes, ya descompone la topología de la red en secciones llamadas “áreas”
* Los enrutadores dentro de un área solo necesitan conocer la topología de área
* Existen distintos tipos de enrutadores
  + Enrutador designado (DR): Encargado de mantener actualizada de la base de datos topológica dentro de un área
  + Enrutador de frontera de área (ABR): Enrutador con interfaces en más de un área (No más de 5 áreas)
  + Enrutador de frontera de sistema autónomo (ASBR): Enrutador con interfaces en más de un A.S.
* Tipos de áreas usadas en OSPF:
  + Área cero (blackbone): Forma el núcleo de una res OSPF. Es la única área que siempre debe estar presente en cualquier red OSPF, ya que mantiene conexión física/lógica con todas las demás áreas. La conexión entre un área y el blackbone se realiza mediante ABR
  + Área stub: No recibe rutas externas (rutas que fueron inyectadas desde otro protocolo de enrutamiento), por lo que se apoya de rutas predeterminadas (último recurso) para poder enviar tráfico a rutas fuera del segmento.
  + Área not-so-stubby: Área tipo stub que puede importar rutas externas de A.S.s y enviarlas al backbone, pero no puede recibir rutas externas de A.S.s desde backbone u otras áreas
* OSPF mantiene actualizada la capacidad de enrutamiento entre entre los nodos de una red mediante la difusión de la topología e información estado-enlace mediante los siguientes tipos de mensaje: (Cada 10 A 30 segundos)
  + Paquetes hello (tipo1): cada enrutador envía periódicamente a sus vecinos un paquete que contiene el listado de vecinos reconocidos por él, indicando el tipo de relación entre ellos.
  + Paquete de descripción de base datos estado-enlace (LSDB): Se emplean en el intercambio de base de datos estado-enlace entre 2 nodos y permiten informar al otro nodo acerca de los siguientes contenidos en la LSDB propia, mediante un resumen de estas
  + Paquetes de estado-enlace (LSA)
    - Tipo 2: Representa un DR para un enlace multiacceso
    - Tipo 3: Representa una sumarización
    - Tipo 4: Representa un ASBR
    - Tipo 5: Representa una ruta externa al dominio OSPF (inyectada por redistribución)
    - Tipo 6: Multicast
    - Tipo 7: Usado en áreas stub en lugar de mensajes tipo 5
  + Mensaje hello



En el caso del tipo de autenticación si

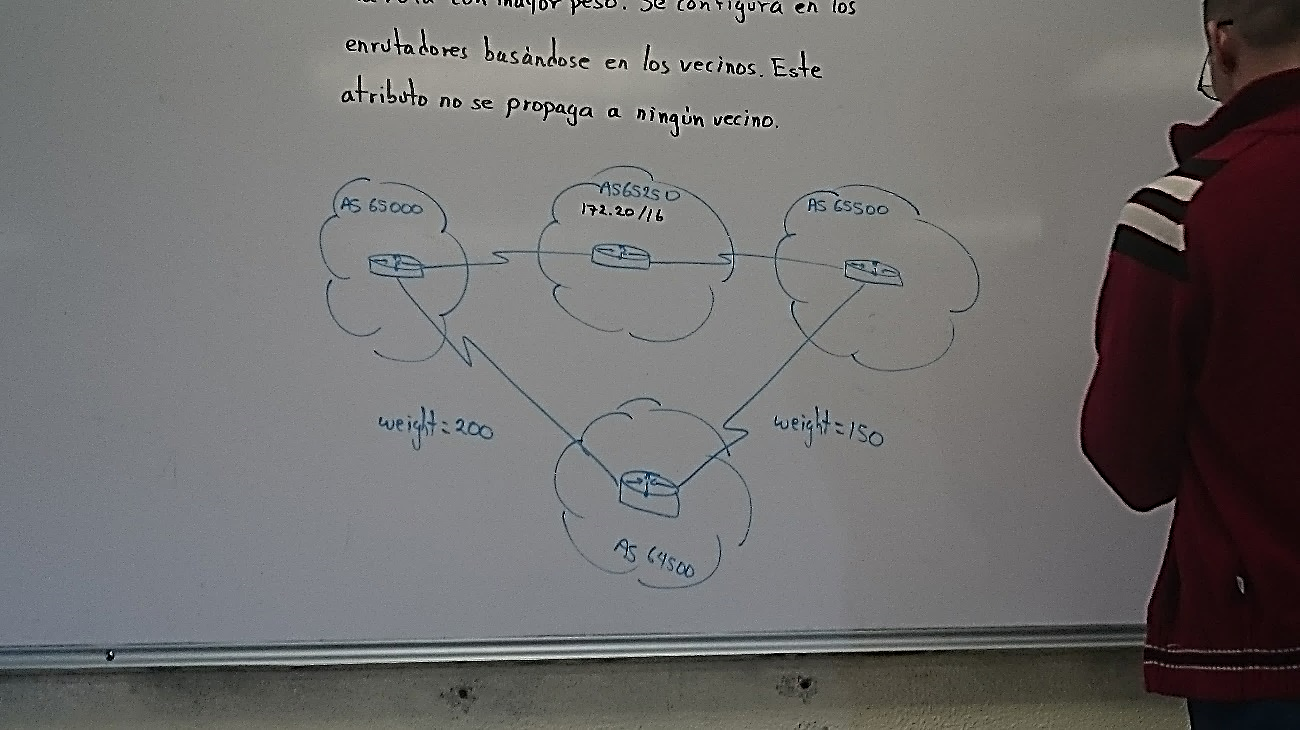
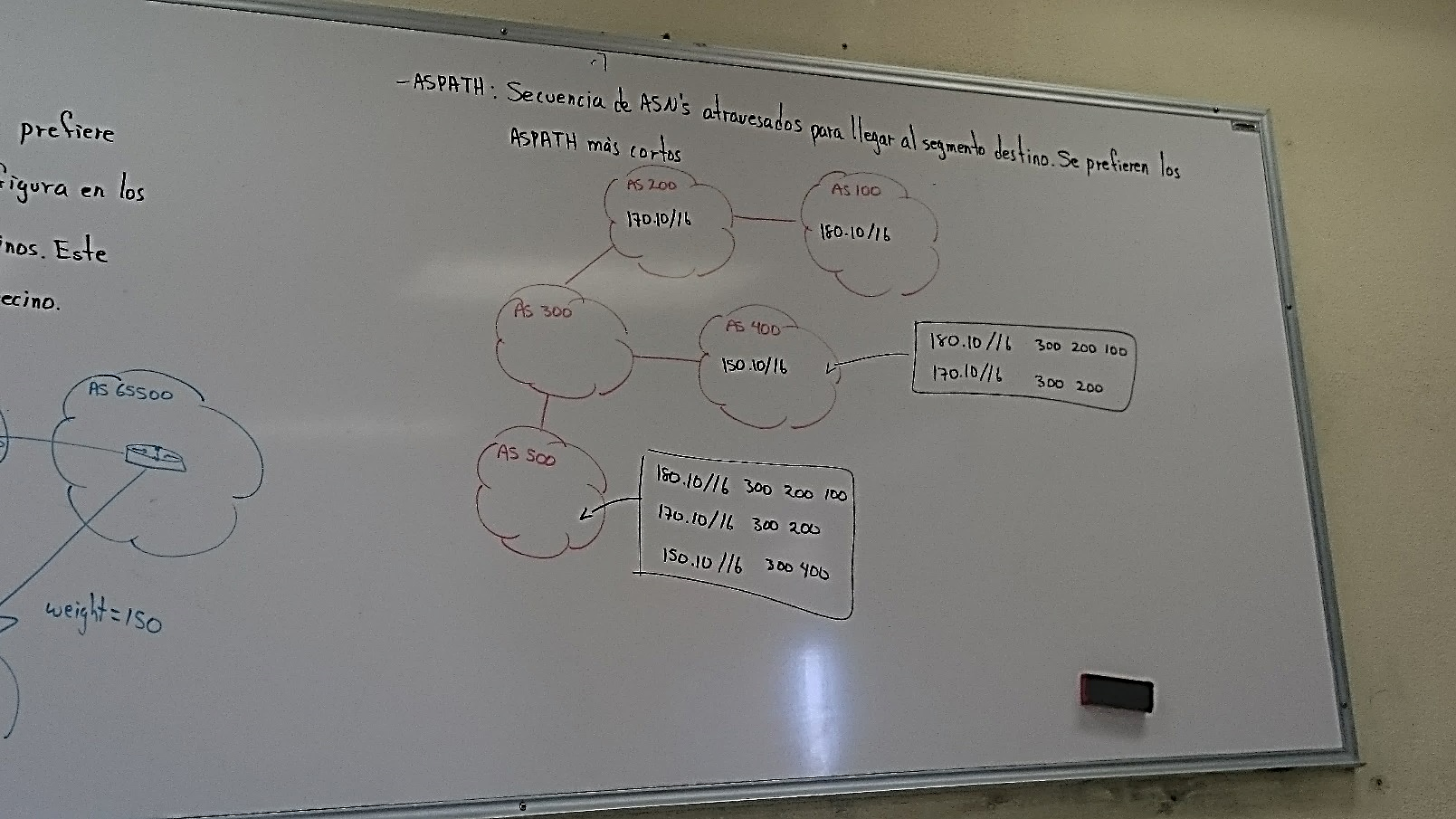
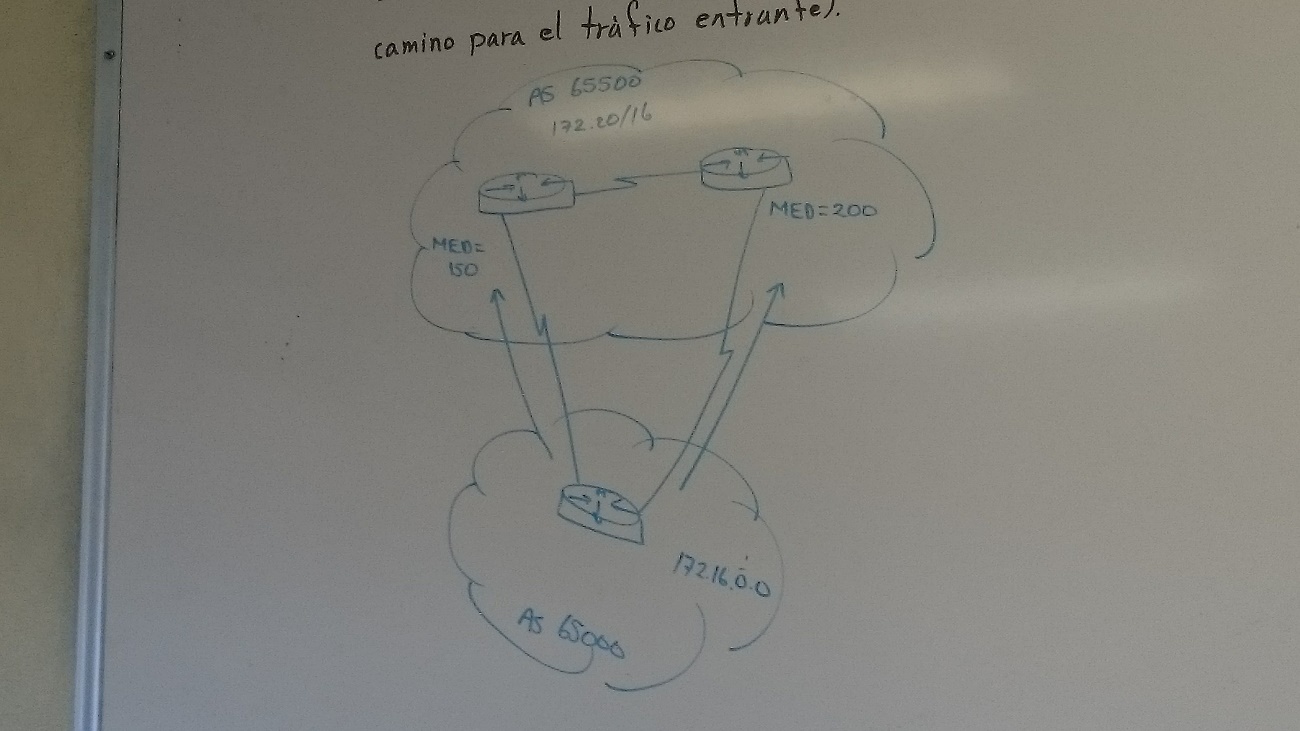
En el caso de las opciones

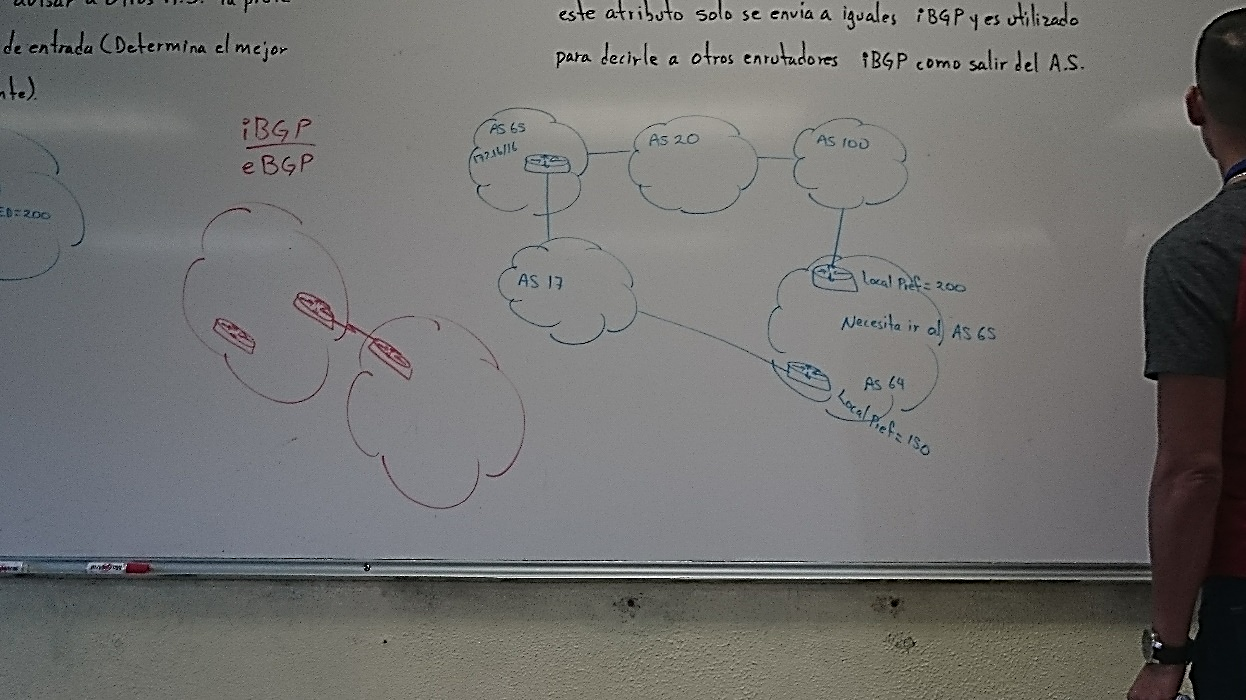
EL INTERVALO DE MUERTE ES 4 VECES EL DE SALUDO

**Estados alcanzados para lograr la adyacencia:**

* Desactivado (down): El proceso OSPF no ha intercambiado información con ningún vecino. OSPF se encuentra en espera de pasar al siguiente estado
* Inicialización (init): Los enrutadores envían paquetes Hello a intervalos regulares (10 a 30 segundos) tratando establecer una relación con posibles vecinos. Cuando una interfaz recibe su primer paquete hello se entra en este estado. Existen 2 tipos de relaciones: Bidireccional y adyacencia
* Bidireccional (2-way): Empleando paquetes hello cada enrutador trata de establecer el estado de comunicación de 2 vías con cada vecino en la misma red. Se entra en este estado al verse a sí mismo en un mensaje de saludo de un vecino
* Intercambio(Exstart): Se utilizan paquetes con descripciones de base de datos, así como mensajes hello para negociar quien tendrá rol de maestro y quien el de esclavo (El enrutador con mayor ID fana el derecho de ser maestro)
* Intercambio(Exchange): Los enrutadores describen su base de datos de estado enlace entre ellos y comparan la información que ya conocían con la recibida. Si se recibe información nueva, el enrutador solicitará una actualización completa a su vecino (estado de carga)
* Carga(Loading): Después que las bases de datos han sido completamente descritas entre vecinos, estos pueden requerir información más completa empleando paquetes de tipo 3 (Requerimientos estado enlace), mismos que serán respondidos con paquetes tipo 4 (LSA)
* Adyacencia completa (Full): Cuando el estado de carga ha sido completado, los enrutadores se vuelen completamente adyacentes.
  + router ospf [número]
  + network [ip] [máscara comodín] area[#]
  + interface loopback [número]
  + ip address [ip] [máscara]
  + bandwith [kbps]
  + ip ospf cost [número]
  + ip ospf hello-interval [segundos]
  + ip ospf dead-interval [segundos]
  + redistribute static
  + área[número] virtual-link [ID enrutador]
  + show ip ospf database
  + show ip ospf interface
  + show ip ospf neighbor
  + show ip route

**BGP (Border Gateway Protocol)**

* Es un protocolo de enrutamiento exterior utilizado para enrutar paquetes entre sistemas autónomos
* Puede ser implantado en ISPs, así como empresas u organizaciones con más de un ISP
* Está definido en el RFC 1441
* Es un protocolo vector distancia, pero diferente a RIP, BGP no anuncia vía broadcast (usa sesiones TCP, puerto 179)
* En BGP los iguales (peers) ya están fijos (no se descubren)
* BGP necesita anunciar distintos atributos de rutas a sus iguales, para que ellos puedan elegir el mejor siguiente salto
* Atributos considerados:
  + Peso (weight): Atributo creado por cisco, se prefiere la ruta con mayor peso. Se configura en los enrutadores basándose en los vecinos. Este tributo no se propaga a ningún vecino. Rutas originadas en el A.S. tienen peso 32468, 0 para las otras
  + **ASPATH:** Secuencia de ASNs atravesados para llegar al segmento destino. Se prefieren los ASPATH más cortos
  + **Discriminador MultiSalida (MED):** El camino con el menor MED es preferido. Usado para avisar a otros A.S. la preferencia relativa de los puntos de entrada (Determina el mejor camino para el tráfico entrante)
  + **Preferencia Local:** Las rutas con mayor preferencia local son preferidas, este atributo solo se envía a iguales iBGP Y es utilizado para decirle a otros enrutadores Ibgp COMO SALIR DEL A.S.



* + Proceso de selección de la mejor ruta basado en los atributos:
    - Primero se prefiere la ruta con el peso más grande
    - Si existe más de una opción, se prefiere la ruta con la preferencia local más grande
    - Si aún existe más de una opción, se prefiere la ruta originada por BGP en el enrutador
    - Si aún existe más de una opción, se prefuere la ruta con el ASPATH más corto
    - Si aún existe más de una opción, se prefiere la ruta con el MED más pequeño
    - Si aún existe más de una opción, se prefiere la ruta con el menor ID de vecino (Router ID)