

Redes de Computadoras
Solución examen – 19 de febrero 2018
(ref: solredes20180219.odt)

Instrucciones

- Indique su nombre completo y número de cédula en cada hoja.
- Numere todas las hojas e indique la cantidad total de hojas que entrega en la primera.
- Escriba las hojas de un solo lado y utilice una caligrafía claramente legible.
- Comience cada pregunta teórica y cada ejercicio en una hoja nueva.
- Sólo se responderán dudas de letra. No se responderán dudas de ningún tipo los últimos 30 minutos del examen.
- El examen es individual y sin material. Apague su teléfono celular mientras esté en el salón del examen.
- Es obligatorio responder correctamente al menos 15 puntos en las preguntas teóricas y 20 de los problemas prácticos. Los puntos ganados en el curso se suman a los puntos de teórico.
- El puntaje mínimo de aprobación es de 60 puntos.
- Para todos los ejercicios, si es necesario, puede suponer que dispone de los tipos de datos básicos (p.ej. lista, cola, archivo, string, etc.) y sus funciones asociadas (ej: tail(lista), crear(archivo), concatenar(string, string).
- Justifique todas sus respuestas.
- Duración: 3 horas. Culminadas las 3 horas el alumno no podrá modificar las hojas a entregar de ninguna forma.

Preguntas Teóricas

Pregunta 1 (6 puntos)

En el contexto de una LAN, suponga que por error dos hosts A y B tienen la misma dirección MAC, están conectados a un HUB y tienen direcciones IP diferentes, IP_A e IP_B respectivamente. Suponga también que en un host C conectado al mismo HUB, con MAC_C e IP_C , se ejecuta el comando ping -c2 IP_A . Asumiendo que las tablas de ARP están vacías al comienzo, muestre el intercambio completo de tramas que ocurriría hasta que el host C recibe el segundo ECHO REPLY. Explique el motivo por el que se transmite cada paquete.

Solución

Dado que tenemos un HUB todas las tramas son replicadas por todas las bocas, así que en todos los casos el host B recibirá las tramas/paquetes pero siempre los descartará porque son enviados a IP_A o porque aunque sean broadcast como en ARP no consultan por IP_B .

Como las tablas ARP están vacías C envía una trama ARP Query a la dirección de broadcast consultando por IP_A usando, usando MAC_C e IP_C

A responde con un ARP Reply comunicando que IP_A se encuentra en MAC_A , y lo hace por unicast a MAC_C .

En este momento las tablas ARP de A y C contienen los mapeos $IP_A - MAC - tiempo$ e $IP_C - MAC_C - tiempo$

C envía un ECHO REQUEST a IP_A

A responde con un ECHO REPLY a IP_C

y vuelve a repetirse los ECHO REQUEST Y REPLY, ya que no es necesario volver a hacer el ARP ya que asumimos que las entradas de la tabla ARP no se vencieron. El host B no participa pues, si bien a nivel de capa 2 hay colisiones, la clasificación de capa 3 descarta los paquetes que no involucran IP_B en las comunicaciones.

Pregunta 2 (8 puntos)

Describe las características del protocolo Neighbor Discovery de IPv6, y señale las diferencias con ARP, utilizado en IPv4. En particular describe los distintos tipos de mensajes utilizados, y cuándo se utiliza broadcast, multicast y unicast.

Solución

Se utiliza NDP para determinar las direcciones de capa de enlace (MAC) de los nodos pertenecientes al mismo segmento de red, y asimismo para encontrar los routers vecinos para SLAAC. NDP emplea la mensajería de ICMPv6, y es la base para permitir el mecanismo de autoconfiguración en IPv6. El protocolo ND define cinco tipos de paquete ICMPv6 para *router solicitation* (RS), *router advertisement* (RA), *neighbor solicitation* (NS), *neighbor advertisement* (NA), y *network redirect* (NR).

En particular, los nodos envían mensajes NS para determinar dinámicamente la correspondencia IPv6 - MAC. NS utiliza *multicast* cuando el nodo necesita resolver una dirección y *unicast* para determinar alcance (*reachability*). De esta forma reemplaza los mensajes "ARP request" de IPv4 (transportados en tramas de *broadcast ethernet*), brindando características mejoradas y teniendo mejor integración con la suite IPv6.

ND forma parte de los mecanismos de base de IPv6, mientras que en IPv4 se utilizan diferentes mecanismos para alcanzar funcionalidades similares: ARP, ICMP router discovery, e ICMP redirect.

Pregunta 3 (8 puntos)

- a) ¿Qué es un Sistema Autónomo (Autonomous System, AS)?
- b) Explique los conceptos Interior Gateway Protocols (IGP) y Exterior Gateway Protocols (EGP) vinculándolos con el concepto explicado en la parte a) y brinde un ejemplo de cada uno.
- c) Describa brevemente el uso que hace BGP (Border Gateway Protocol) del atributo AS-PATH.

Solución

Capítulo 4 del libro. Capa de Aplicación. Diapos 96, 97, 104, 118..122 del material de referencia del teórico.

- a) AS es un concepto vinculado al enrutamiento jerárquico. Si podemos decir que Internet es una "red de redes" y se pretende que en la gestión de ellas mantenga un orden, debemos identificar redes con autonomía administrativa para el conjunto de dispositivos que la componen, y en particular para los protocolos de enrutamiento configurados en ella. Por lo tanto podemos decir que un AS es *una agrupación de routers con una administración común*.
- b) Interior Gateway Protocols (IGP): protocolos de enrutamiento que "corren" dentro de un AS (routing intra-AS); ejemplos: RIP y OSPF. Exterior Gateway Protocols (EGP): protocolos de enrutamiento que "corren" entre ASs (routing inter-AS); ejemplo: BGP.
- c) El atributo AS-PATH contiene la lista de ASs que ha atravesado la publicación de un prefijo. Cuando un prefijo es pasado a un AS, éste agrega su ASNumber al atributo. Este atributo tiene dos usos fundamentales: evitado de loops en las publicaciones (si un AS se encuentra a sí mismo en el AS-PATH) y, si es tenido en cuenta en el proceso de selección de ruta y cuando se lo considere (existen otros atributos), se seleccionará aquella que tenga el AS-PATH más corto.

Pregunta 4 (10 puntos)

Considere un usuario situado en un equipo H (con dirección 10.10.10.20) el cual se encuentra en una red local que provee salida a Internet a través de un gateway con dirección 10.10.10.1 y cuenta con un servidor DNS local en la dirección 10.10.10.10.

El usuario pretende descargar desde Internet utilizando un navegador web la página <http://www.empresa.com.uy/index.html>. Esta página cuenta con dos enlaces a las fotos /images/foto1.png e /images/foto2.png, alojados en el mismo servidor web que atiende peticiones en el puerto 80 TCP.

Se pide:

Indique las sesiones TCP y UDP vistas en la red local (10.10.10.0/24), especificando claramente direcciones, puertos y para que se utiliza cada sesión. Suponga que el navegador web utiliza HTTP 1.1, que el servicio DNS y el equipo H no conocen la dirección IP de www.empresa.com.uy, y que el servidor DNS genera consultas iterativas.

Solución

Protocolo	IP Origen	IP Destino	Puerto Origen	Puerto Destino	Datos
UDP	10.10.10.20	10.10.10.10	14000	53	Consulta DNS de tipo A por www.empresa.com.uy . Genera consulta y espera resultado en puerto 14000. Consiste de una solicitud y una respuesta.
UDP	10.10.10.10	DNS raíz	20000	53	Consulta DNS de tipo A por www.empresa.com.uy . Devuelve el RR NS autoritativo del TLD uy y el RR A correspondiente (en cada caso, puede ser más de uno). Consiste de una solicitud y una respuesta.
UDP	10.10.10.10	IP_TLD	20001	53	Consulta DNS de tipo A por www.empresa.com.uy . Devuelve el RR NS autoritativo de com.uy (en cada caso, puede ser más de uno). Consiste de una solicitud y una respuesta.
UDP	10.10.10.10	IP_DNS_Aut_ "com.uy"	20002	53	Consulta DNS de tipo A por www.empresa.com.uy . Devuelve el RR NS autoritativo de com.uy y el RR A correspondiente (en cada caso, puede ser más de uno). Consiste de una solicitud y una respuesta.
UDP	10.10.10.10	IP_DNS_Aut_ "empresa.com.uy"	20003	53	Consulta DNS de tipo A por www.empresa.com.uy . Devuelve IP (RR del tipo A) de www.empresa.com.uy (167.6.6.20). Consiste de una solicitud y una respuesta.
TCP	10.10.10.20	167.6.6.20	14001	80	Conexión que primero solicita la página index.html, a través de un HTTP GET. La respuesta contiene: HTTP 200 OK y la página index.html solicitada. A continuación solicitando la imagen foto1.png, a través de un HTTP GET. La respuesta contiene: HTTP 200 OK y la imagen foto1.png. Por último solicita a imagen foto2.png, a través de un HTTP GET. La respuesta contiene: HTTP 200 OK y la imagen foto2.png.

Pregunta 5 (8 puntos)

En el protocolo de capa de transporte TCP:

- a) ¿Por qué se necesitan números de secuencia?
- b) ¿Por qué son necesarios temporizadores?
- c) En una red con tasas de errores ínfimas y recursos sobredimensionados, ¿utilizaría TCP?

Solución

a) TCP es un protocolo de transporte que asegura que no se perderán datos durante la transmisión. Para ello, el emisor debe recibir confirmación de que el receptor recibió correctamente cada uno de los paquetes enviados.

Para aprovechar mejor el enlace, TCP es también un protocolo pipelined por lo que el emisor envía varios paquetes antes de recibir confirmación por alguno de ellos. Por esta razón, el emisor debe poder distinguir a que paquete o paquetes corresponde una determinada confirmación recibida por lo que numera los paquetes enviados secuencialmente. Las confirmaciones enviadas por el receptor utilizan esos números para distinguir que paquete o conjunto de paquetes están confirmando.

b) TCP es un protocolo de transporte que asegura que no se perderán datos durante la transmisión. Para ello, el emisor debe recibir confirmación de que el receptor recibió correctamente cada uno de los paquetes enviados. De no recibir esta confirmación, el emisor no sabe si esto se debe a que se perdió el paquete enviado, la confirmación o simplemente alguno de los dos esta demorado. La única solución es retransmitir el paquete no confirmado. Para ésta retransmisión debe esperar un tiempo prudencial luego de enviado cada paquete por lo que necesita un timer que le indique cuando hacerlo.

c) Una hipotética red con las condiciones mencionadas -tasas de errores ínfimas y recursos sobredimensionados- no está exenta de otras que pueden afectar seriamente a determinadas aplicaciones y/o servicios; a modo de ejemplo podemos citar que dicha red puede ofrecer diversos caminos entre un origen y un destino. En dicho escenario, casi cualquier implementación de e-banking o de comunicaciones asociadas a infraestructuras críticas, requiere de un protocolo de transporte como TCP, ya que las funcionalidades descritas en las partes a) y b) resultan imprescindibles para reducir los riesgos de acciones incorrectas que se puedan tomar a partir de inferir la ocurrencia -o no- de algún mensaje.

Redes de Computadoras

Problemas Prácticos

Problema 1 (30 puntos)

Se considera una aplicación web desplegada en N centros de cómputos distribuidos por el mundo, la cual es accedida mediante la URL `http://aplicacionweb.com`. Las sesiones de los usuarios son atendidas por alguno de los servidores distribuidos, y para mejorar el tiempo de respuesta se redirigen las sesiones al servidor geolocalizado a menor distancia. Se utiliza el mecanismo de redirección a través de HTTP, debiendo responder a la solicitud con el encabezado *meta*, mostrado a continuación, donde `URL_DESTINO` es la dirección web de un servidor que lo debe atender:

```
<head>
  <meta http-equiv="refresh" content="0; URL=URL_DESTINO" />
</head>
```

Se cuenta además con la función *nearest_server(IP)*, que dada la IP pasada por parámetro, devuelve la URL del servidor con menor latencia a la misma.

Se pide:

- Implemente en alto nivel el servidor que atiende la invocación del navegador, (a través de `http://aplicacionweb.com`) y lo redirija al servidor más cercano, especificando detalladamente el uso de sockets en su implementación.
- Proponga algún mecanismo para implementar la función *nearest_server(IP)* (no se piden implementaciones, solo descripciones de alto nivel). Para la solución propuesta especifique que condiciones deben cumplirse en referencia a los nombres de dominio utilizados por las URL alojadas en los distintos centros de cómputo. Aporte ejemplos que aclaren su propuesta.

Solución

a)

```
//Funcion de atencion al cliente
void *client_process(void *);

//Funcion principal
int main(int argc , char *argv[]){
    int socket_srv , client_sock , c , *new_sock;
    struct sockaddr_in server , client;

    //Creacion del socket
    socket_srv = socket(AF_INET , SOCK_STREAM , 0);
    if (socket_srv == -1){
        printf("Error en creacion del socket");
    }

    //Configuracion del socket
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = INADDR_ANY;
    server.sin_port = htons( 3000 );

    //Bind del socket
    if( bind(socket_srv, (struct sockaddr *)&server , sizeof(server)) < 0){
        printf("Error en bind");
        exit 1;
    }

    //Listen seteo para atencion del server
    listen(socket_srv , 3);

    //Acceptando conexiones entrentes
    puts("Esperando conexiones...");
    c = sizeof(struct sockaddr_in);

    c=sizeof(struct sockaddr_in);
    while(client_sock=accept(socket_srv, (struct sockaddr*)&client, (socklen_t*)&c)){
        puts("Conexion aceptada");
        pthread_t sniffer_thread;
        new_sock = malloc(1);
        *new_sock = client_sock;
        // tomo IP del cliente
        cli_IP = inet_ntoa(c.sin_addr);
        // invoco el thread
```

Redes de Computadoras

```
if( pthread_create( &sniffer_thread ,NULL,client_process,new_sock,cli_IP) < 0){
    printf("Error al atender el thread de atencion");
    exit 1;
}
}

if (client_sock < 0){
    printf("Error en accept");
    exit 1;
}
exit 0;
}

/*
Proceso que maneja la conexion con cada cliente
*/
void *client_process(void *socket_srv, char * cli_IP){
    //Obtiene descriptor del socket
    int sock = *(int*)socket_srv;
    int n;
    char URL_DEST[100], send_buff[2000], cli_buff[2000];

    while((n=recv(sock,cli_buff,2000,0))>0){
        //analizo si es GET a la URL especificada
        if(contains(cli_buff,"GET /") and contains(cli_buff,"Host: aplicacionweb.com") ){
            URL_DEST = nearest_server(cli_IP)
            send_buff=getMsgRedirect(URL_DEST);
        }else{
            send_buff=ErrorMessage();
        }
        send(sock,send_buff,n,0);
    }
    close(sock);

    if(n==0) {
        printf("Cliente Desconecto");
    }else{
        printf("Error en recepcion");
    }
    return 0;
}
}
```

b)

Para implementar la función *nearest_server(IP)* se propone consultar todos los centros de computos, solicitando realicen un ping a la IP pasada por parámetro y devuelvan el valor de RTT obtenido. Para todos los valores obtenidos, se toma el menor de ellos.

El proceso a ejecutar es el siguiente:

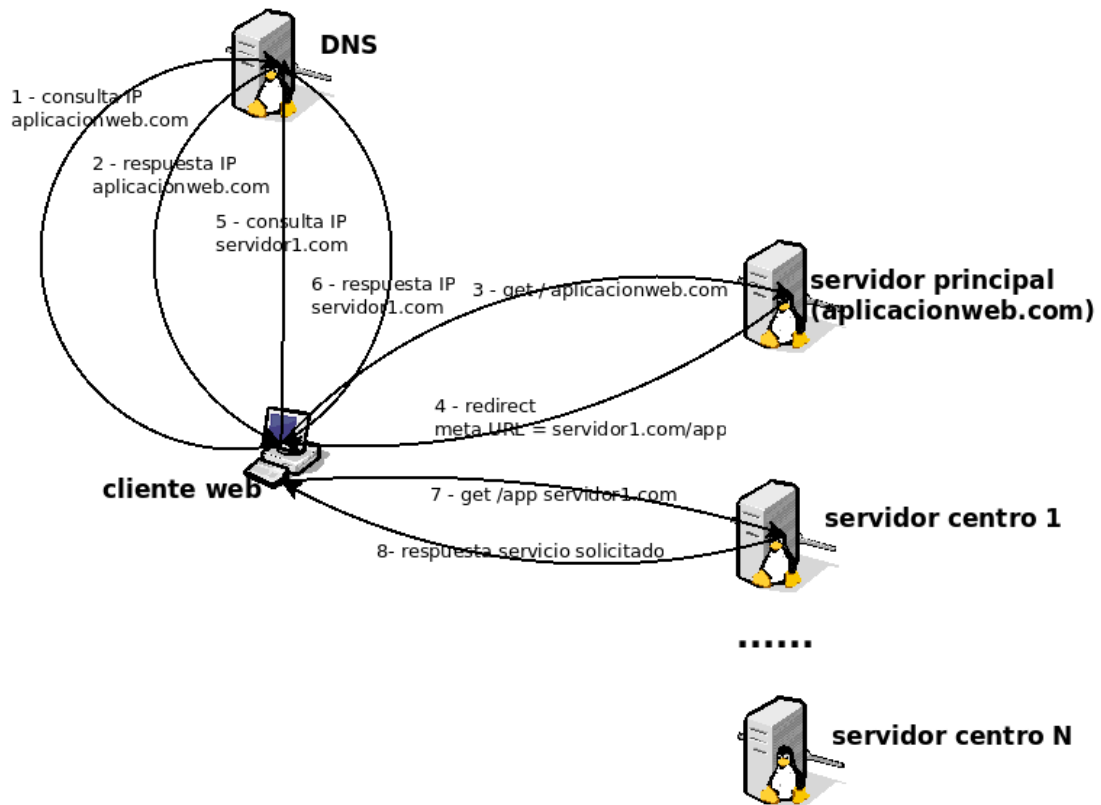
```
for dc in dataCenter
do
    RTT(dc)=getRTT(dc,IP)
done
rtt_min = min(RTT(i)) // tomo valor minimo de los obtenidos.
dc = getDC(rtt_min) // devuelve el dc con RTT igual al pasado por parámetro
return dc
```

Los nombres de dominios utilizados por los diferentes *centros de cómputos* deben ser diferentes, para asegurarse que el DNS responde la dirección IP del servicio brindado por el centro de cómputos definido por la función *nearest_server()*.

La complejidad que trae éste problema es que las aplicaciones que corren en los diferentes centros de computos, deben tener la posibilidad de configurar las llamadas a las páginas con el nombre de dominio absoluto utilizado por él, dado que si utilizase nombres genéricos ocurre que puede cambiar la atención hacia otro centro de cómputos, en la ejecución de la aplicación.

Para visualizar con un ejemplo, se considera el caso en que la redirección (respuesta de *nearest_server*), es hacia el *servidor1.com*, obteniendo el resultado deseado. En la imagen se muestran los 8 pasos requeridos para la solución del problema.

Redes de Computadoras



Como puede verse, en los 8 pasos presentados, si los servidores tuviesen el mismo nombre de dominio, el DNS respondería uno de estos, sin identificar cual. Además ocurriría que si el DNS respondiese la IP de todos, sería el cliente el que elije a cual ir, y no sería redireccionado al centro de cómputo deseado, respuesta de `nearest_server()`.

Problema 2 (30 puntos)

Una empresa contrata una conexión a Internet a un proveedor que instala un router con una configuración que deja la red pública 190.1.1.0/29 accesible en la interfaz de red local de dicho router con dirección IP 190.1.1.1. Dicho router es administrado por el proveedor de Internet (es decir, su configuración NO se puede cambiar), y permite acceso completo desde/hacia Internet a la red pública. La empresa tiene un servidor DNS, y además pretende tener un servicio web distribuido, balanceando carga entre dos o más servidores físicos. La empresa está organizada en las secciones Ventas, Administración y Gerencia, que tienen 20, 30 y 10 usuarios respectivamente, pero es posible que crezcan un 60% c/u. Todos los usuarios deben tener acceso a Internet y a los servidores públicos de la empresa, pero no deben poder acceder al resto de las secciones.

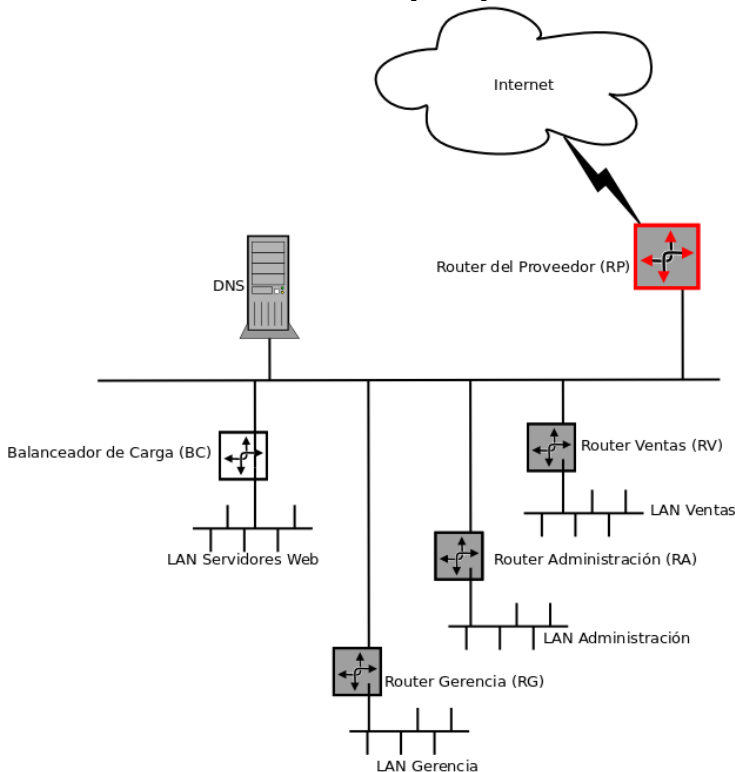
Se pide:

- a) Proponga una arquitectura de red para cumplir con los requerimientos. Especifique los dispositivos a agregar y las funcionalidades que deben tener. Presente la solución propuesta en un diagrama.
- b) Asigne direcciones IP para todos los sistemas de la empresa, utilizando direcciones IP privadas en caso de ser necesario. La asignación debe ajustarse estrictamente a las necesidades especificadas.
- c) Configure las tablas de forwarding de todos los dispositivos para cumplir con los requerimientos. Utilice el formato Prefijo de Destino | Interfaz de salida | Next-Hop.

Solución

a)
Se presenta la propuesta de solución de la figura:

- El DNS es un servidor que implementa el servicio de nombres.



- Los routers RV, RA y RG deben implementar NAT para cada una de las subredes (Ventas, Administración, Gerencia); las tablas de forwarding deben configurarse adecuadamente para cumplir con los requerimientos (ver parte c).
- El balanceador de carga BC debe tener algún algoritmo de balanceo (por ejemplo algo simple como round-robin), y debe mantener las sesiones entre los usuarios y el servidor elegido de forma transparente al usuario.

b)
 LAN pública 190.1.1.0/29
 RP: 190.1.1.1
 DNS: 190.1.1.2
 BC: 190.1.1.3
 RV: 190.1.1.4
 RA: 190.1.1.5
 RG: 190.1.1.6

LAN Ventas: 20*1.6 -> 32 hosts
 Se necesitan además: 1 dir IP para el RV, dirección de red y broadcast, 35 direcciones IP en total -> prefijo /26

Redes de Computadoras

LAN Administración: 30*1.6 -> 48 hosts

Se necesitan además: 1 dir IP para el RA, dirección de red y broadcast, 51 direcciones IP en total -> prefijo /26

LAN Gerencia: 10*1.6 -> 16 hosts

Se necesitan además: 1 dir IP para el RG, dirección de red y broadcast, 19 direcciones IP en total -> prefijo /27

Dado que las tres subredes son estancas (es decir, solo se comunican con "el exterior" mediante NAT pero no entre si), se puede usar el mismo prefijo de red, por ejemplo se puede asignar:

LAN Ventas: 192.168.1.0/26

LAN Administración: 192.168.1.0/26

LAN Gerencia: 192.168.1.0/27

Para evitar confusiones y permitir futuros cambios de configuración se pueden asignar prefijos diferentes, por ejemplo:

LAN Ventas: 192.168.1.0/26

LAN Administración: 192.168.2.0/26

LAN Gerencia: 192.168.3.0/27

Para la LAN de Servidores Web se puede utilizar cualquier prefijo porque no se imponen restricciones, por ejemplo 10.0.0.0/8

En todos los casos se asigna la dirección de la LAN de direcciones privadas .1 a la interfaz de los routers y el balanceador.

c)

DNS:

Prefijo de Destino	Interfaz de salida	Next-Hop
190.1.1.0/29	eth0 (pública)	Directamente conectado
0.0.0.0/0	eth0 (pública)	190.1.1.1

BC:

Prefijo de Destino	Interfaz de salida	Next-Hop
190.1.1.0/29	eth0 (pública)	Directamente conectado
0.0.0.0/0	eth0 (pública)	190.1.1.1
10.0.0.0/8	eth1 (privada)	Directamente conectado

Routers de subredes Ventas, Administración y Gerencia: solo deben tener configuradas las redes directamente conectadas y la ruta por defecto. NO se deben configurar rutas a las otras subredes; por este motivo se utilizan tres routers y no uno solo, ya que en ese caso no se podría resolver mediante el enrutamiento la asilación entre subredes.

RV:

Prefijo de Destino	Interfaz de salida	Next-Hop
190.1.1.0/29	eth0 (pública)	Directamente conectado
0.0.0.0/0	eth0 (pública)	190.1.1.1
192.168.1.0/26	eth1 (privada)	Directamente conectado

RA:

Prefijo de Destino	Interfaz de salida	Next-Hop
190.1.1.0/29	eth0 (pública)	Directamente conectado
0.0.0.0/0	eth0 (pública)	190.1.1.1
192.168.2.0/26	eth1 (privada)	Directamente conectado

RG:

Prefijo de Destino	Interfaz de salida	Next-Hop
190.1.1.0/29	eth0 (pública)	Directamente conectado
0.0.0.0/0	eth0 (pública)	190.1.1.1
192.168.3.0/27	eth1 (privada)	Directamente conectado

Redes de Computadoras

Todos los sistemas de las subredes RV, RA, RG y los servidores web tendrán en su tabla de forwarding una entrada de la subred local directamente conectada, y una ruta por defecto cuyo next-hop es la dirección IP .1 de cada subred. Todos estos sistemas tienen una sola interfaz de red (por ejemplo eth0).