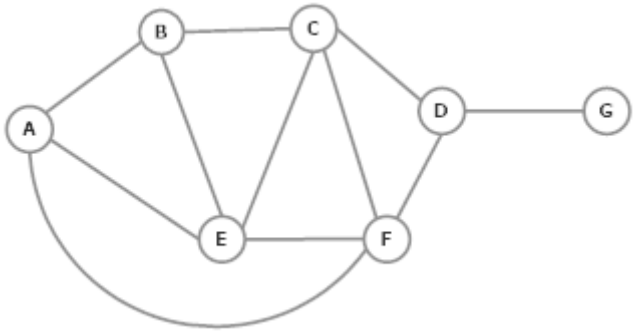
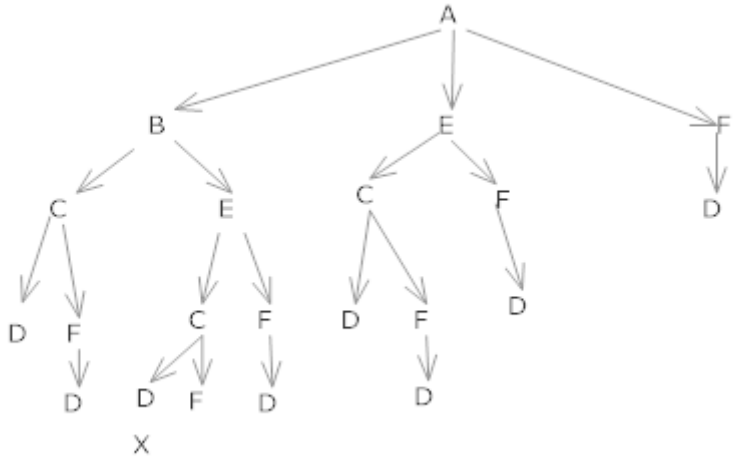
Ej. 4): Asumimos que se tiene la subred de la figura de abajo. Se desea enviar un paquete del nodo A al nodo D usando inundación.

- Se cuenta la transmisión de un paquete a lo largo de una línea como una carga de uno
- Estudiar del libro o las filminas los algoritmos de inundación de conteo de saltos y de inundación selectiva.
- ¿cuál es la carga total generada si se usa inundación selectiva, un campo de conteo de saltos es usado y es inicialmente fijado en 4?



Ayuda: construir árbol de envío de paquetes.



¿Cuál es entonces la carga de la red?

La carga total es de 20.

Ej. 8): Supongamos que tenemos una subred con forma de anillo (ciclo) de *N* enrutadores (i.e. cada enrutador está conectado con 2 enrutadores vecinos) y que se usa el protocolo de estado de enlace;

- cada enrutador tiene dos líneas con un vecino: una para enviar y una para recibir;
- suponiendo que un paquete que atraviesa una línea se cuenta como una carga de 1;
- ¿cuál es la carga total en la subred para el proceso entero para la actualización de las tablas de enrutamiento?

Ayuda: hacer las cuentas para cada etapa del algoritmo de enrutamiento.

Carga de ver los vecinos:

2*N paquetes Hello + 2 * N respuestas a Hello = 4 N

Obtención de retardos a los vecinos

2 * N paquetes Echo + 2 * N respuestas a Echo = 4 N

Inundación de la red con paquetes de estado de enlace

El paquete de cada enrutador recorre N. Como hay N enrutadores es N *N la carga.

La carga total del algoritmo va a ser de: 8 N + N*N

Ej. 12): Enrutamiento jerárquico: asumir que todos los elementos de nivel *n* contienen la misma cantidad de elementos de nivel *n+1*. Suponga que hay 3 niveles. Responder:

1. ¿cuántos enrutadores hay en la subred? Dar una fórmula. Justificarla.
2. Explicar cómo se asignan nombres a elementos de nivel 1, a elementos de nivel 2 y a elementos de nivel 3.
3. ¿cuál es el tamaño de las tablas de enrutamiento? Dar una fórmula. Justificarla.

Dividimos la red en regiones, las regiones en clústeres y los clústeres en enrutadores.

Supongamos que hay N regiones, M clústeres por región y K enrutadores por clúster.

Cantidad de enrutadores = N * M * K

Nombres de región = 1,…, N

Nombres de clústeres = <i, j> i: 1..N, j: 1..M

Nombres de enrutadores = <i, j, k> i: 1..N, j: 1..M, k : 1..K

Tamaño de tablas de enrutamiento = cantidad de enrutadores de mi cluster + cantidad de clústeres de mi región + cantidad de regiones – 1 = K + (M-1) + (N -1)

Ej. 13): Supongamos que se tienen 800 enrutadores y se usa el esquema de enrutamiento jerárquico con dos niveles solamente. Suponer que todas las regiones tienen el mismo número de enrutadores. Resolver:

1. ¿Cuántas regiones conviene tener de modo que la tabla de enrutamiento sea lo más chica posible? Justificar la respuesta.
2. Suponer que las distancias se miden como el número de saltos. ¿Usando la respuesta a la pregunta anterior calcular cuál es la cantidad de memoria en total utilizada por la subred necesaria para todas las tablas de enrutamiento? Justifique su respuesta.

Ayuda: expresar número de enrutadores como fórmula en términos de cantidad de regiones y cantidad de enrutadores por región. Luego descomponer en primos 800 y responder 1.

800 = Cantidad de enrutadores = número de regiones * número de enrutadores por región = R * E

800 = 5 * 5 * 2 *2*2*2*2

T = Tamaño de tabla de reenvio es = (R – 1) + E

R = 25 y E = 32, T = 56

R = 32 y E = 25, T = 56

R = 5 *2*2 =20 E = 5*2*2*2= 40 T = 59

La respuesta va a ser:

R = 25 y E = 32, T = 56

R = 32 y E = 25, T = 56

Segunda pregunta:

Hay 800 tablas de reenvio y cada una tiene (R-1) + E entradas.

Entrada = destino, línea de salida, retardo = 10 b + 10 b + 10 b = 30 b

Consideremos R = 25 y E = 32

Memoria total de todas las tablas de reenvio = 800 * 56 * 30 b = 5600 * 30 B

Ej. 16): Indicar qué algoritmo de control de congestión (de los que están en el apunte o en el libro) para capa de red es el más conveniente para cada una de las siguientes situaciones:

1. El buffer de la línea de salida está lleno.
2. La espera para que un paquete sea reenviado por una línea de salida *L* de un enrutador es demasiada y está creciendo (aunque aun hay bastante espacio de buffer); además todos los caminos que unen hosts pasando por *L* son cortos (pocos saltos).
3. La ruta *P* entre un host de origen y un host de destino contiene enrutador *R*; En *P* el enrutador *R* está muy muy lejos del host de origen (muchísimos saltos) y la línea de salida de *R* en *P* se está congestionando muy rápidamente.

Para 1. Si el bufer está lleno se van a perder paquetes indiscriminadamente, a menos que se use desprendimiento de carga. Se podría usar por ejemplo RED.

Para 2. Bit de advertencia va a demorar en el mejor caso un RTT en llegar al origen.

Paquetes reguladores en el peor caso va a demorar algo menos que RTT/2 en llegar el aviso de congestion al origen.

Conviene paquetes reguladores.

Para 3: conviene usar paquetes reguladores salto por salto. Asumimos que todavía no hace falta usar desprendimiento de carga; asumimos que la situación del búfer no es preocupante como para usar desprendimiento de carga.

Ej. 14): ¿Qué ventajas ofrece el algoritmo de paquetes reguladores frente al de bit de advertencia? Dar al menos dos de ellas. La respuesta debe ser justificada.

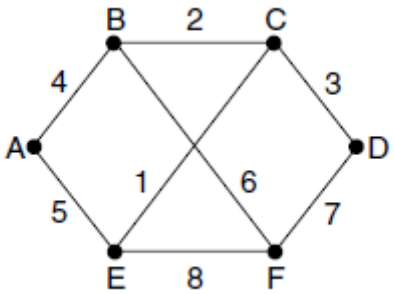
El aviso de congestión es más rápido en paquetes reguladores que en bit de advertencia.

Los paquetes reguladores dicen cuánto hay que regular el tráfico; bit de advertencia no ayuda a regular el tráfico. Además con paquetes reguladores no hace falta usar arranque lento para adivinar cuanto soporta la red, porque los paquetes reguladores me van diciendo lo que el host debe hacer.

Tarea: estudiar vector de distancia.

Segunda parte de ejercitación

Ej. 6): Considerar la subred de la siguiente figura. Se usa enrutamiento de vector de distancia y los siguientes vectores han llegado al enrutador C: desde *B*: (5, 0, 8, 12, 6, 2); desde *D*: (16, 12, 6, 0, 9, 10); y desde *E*: (7, 6, 3, 9, 0, 4). El costo de los enlaces de C a B, D y E son: 6, 3, y 5 respectivamente. ¿Cuál es la nueva tabla de enrutamiento de C? Dar tanto la línea de salida como el costo.



Primero computamos los costos hacia el destino si vamos por un vecino particular

	Por B	Por D	Por E
A	6 + 5 = 11	3 + 16 = 19	5 + 7 = 12
B	6	3 + 12 = 15	5 + 6 = 11
C	-	-	-
D	6 + 12 = 18	3	5 + 9 = 14
E	6 + 6 = 12	3 + 9 = 12	5
F	6 + 2 = 8	3 + 10 = 13	5 + 4 = 9

Para computar la tabla de reenvío para cada destino elegimos el vecino que nos da el menor costo al destino.

destino	Línea de salida	Estimación del costo
A	B	11
B	B	6
C	-	0
D	D	3
E	E	5
F	B	8

Ejercicio C: Supongamos que una empresa tiene un número de IP 180.20.35.115 y que usa NAT con una red interna de prefijo 192.168.0.0/16. Supongamos que por el momento hay solo dos máquinas en la red de la empresa con direcciones IP: 192.168.0.2 y 192.168.0.4. Suponer que existen las siguientes conexiones TCP:

1. (192.168.0.2, 5000) con (198.60.42.12, 80)
2. (192.168.0.2, 2000) con (194.24.0.5, 110)
3. (192.168.0.4, 5000) con (198.60.100.12, 80)

Se pide:

1. Construir la tabla de la caja NAT.
   Luego usar la tabla de la caja NAT construida para responder a las siguientes preguntas:
2. Si sale un mensaje de 192.168.0.2, 5000 hacia 198.60.42.12, 80: ¿Cuál es la traducción del puerto de origen e IP de origen en ese paquete que hace la caja NAT antes de colocar en internet el paquete?
3. Si llegara a la caja NAT un mensaje desde 194.24.0.5, 110, ¿qué IP y puerto de origen tiene ese mensaje que llega a la caja NAT y a qué valores los traduce la caja NAT a esos campos antes de poner el menaje en la red de la empresa?

Tabla de la caja NAT:

Indice 1: <192.168.0.2, 5000>

Indice 2: <192.168.0.2, 2000>

Indice 3: <192.168.0.4, 5000>

Si sale un mensaje de 192.168.0.2, 5000 hacia 198.60.42.12, 80:

192.168.0.2, 5000 está en el índice 1 de la caja NAT.

La compañía tiene IP: 180.20.35.115

Luego la traducción del origen del paquete (IP y puerto) es: <180.20.35.115, 1>

Si llega a la caja NAT un mensaje desde 194.24.0.5, 110 de acuerdo a las conexiones que tenemos, eso quiere decir que el mensaje tiene origen: <180.20.35.115, 2>, porque el índice 2 es el único que me permite acceder a la información de la conexión dentro de la organización.

Le llega a la caja NAT <180.20.35.115, 2> esto se traduce a: <192.168.0.2, 2000> porque es la información que tenemos en el índice 2 de la tabla.

Ejercicio 23: Suponer que un host *A* está conectado a un enrutador *R1*, *R1* está conectado a otro enrutador *R2*, y *R2* está conectado a un host *B*. Suponer que un mensaje TCP que contiene 900 B de datos y 20 B de encabezado TCP es pasado a un código IP en el host A para entregar a B. Mostrar los campos *longitud total*, *identificación*, *DF*, *MF* y *desplazamiento de fragmento* del encabezado IP en cada paquete transmitido sobre los 3 enlaces. Asumir que el enlace *A-R1* puede soportar un tamaño máximo de trama de 1024 B incluyendo un encabezado de trama de 14 B, el enlace *R1-R2* puede soportar un tamaño de trama máximo de 512 B incluyendo un encabezado de trama de 8B y el enlace *R2-B* puede soportar un tamaño de trama máxima de 512 B incluyendo un encabezado de trama de 12 B.

Mensaje TCP 900 B datos y 20 B encabezado = 920 B
Mensaje IP tiene header de 20 B

Para ir de A a R1:

Tengo largo 920 + 20 + 14 = 954 B para trama. Luego cabe perfectamente mensaje en una trama.
Longitud total = 940 en binario.
Identificación tiene 16 bits
DF = 0
MF = 0
Fragment offset = 0

Para ir de R1 a R2:

Como la trama máxima es de 512 B, el mensaje de 900 B de datos no cabe en una trama.

Si lo divido mensaje en 2 mensajes:

464 + 20 + 20 + 8 = 512 B.
484 B de datos en fragmento
484 se pasa y no es múltiplo de 8 (el paquete IP ocupa 484 B).
Luego debe ser 480 B de datos el fragmento (480/8 = 60).
Con esto la suma de todas las piezas me da: 508 B.
Notar que consume 460 B de datos de segmento TCP que se fragmentó.

Luego el segundo fragmento tiene 440 B de datos (para que sume 900 B).
Con esto la suma de todas las piezas me da: 440 + 20 + 20 + 8 = 488 B.

Resumiendo:
Longitud total primer fragmento = 500 (encabezado + datos)
DF = 0

MF = 1
Fragment offset = 0

Longitud total Segundo fragmento = 480 (encabezado + datos)
DF = 0
MF = 0
Fragment offset = 60 (= 480/8)

Para ir de R2 a B:

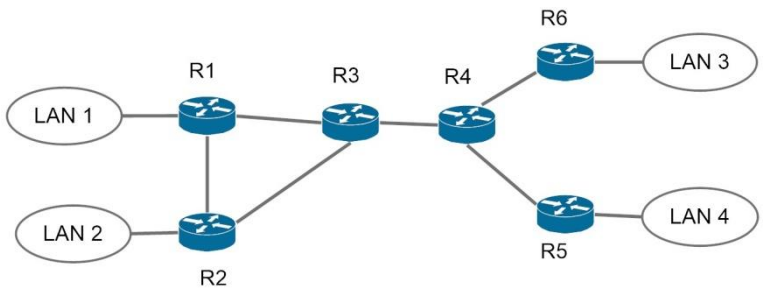
Se soportan paquetes de hasta 500 B porque encabezado de trama es de 12 B.
La subdivisión anterior anda bien y no ocurre más fragmentación. Obtengo de nuevo lo mismo.

Longitud total = 500 en binario.
DF = 0
MF = 1
Fragment offset = 0

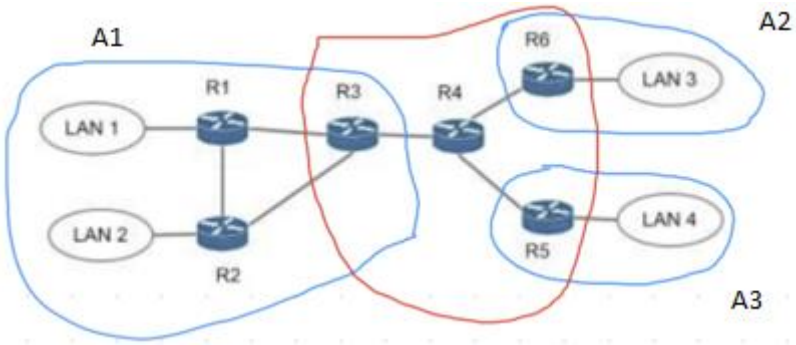
Longitud total = 480 en binario.
DF = 0
MF = 0
Fragment offset = 60

Ejercicio F: Considerar el sistema autónomo de la figura de abajo; asumir que se trabaja con OSPF. Los enrutadores R3, R6, y R5 son de borde de área y todos pertenecen a áreas diferentes. Por simplicidad asumir que cada enlace tiene costo 1 en ambas direcciones. Se pide:

1. Indicar contenido de paquetes de resumen de otras áreas que recibe R2.
2. Construir contenido de avisos de estado de enlace de R3 para enviar a la red dorsal, y a los enrutadores R1 y R2.
3. Construir el grafo que calcula R3 al cuál aplica el algoritmo de Dijkstra.
4. Hacer lo mismo para R2.



Primero hay que determinar las áreas del SA:



Contenido de paquetes de resumen de otras áreas que recibe R2.

Área de izquierda: A1

Área superior derecha: A2

Área inferior derecha es A3

R2 recibe de A2: (LAN 3, 1)

R2 recibe de A3: (LAN 4, 1))

R2 recibe del área dorsal:

(R6, R3) con costo 2

(R3, R6) con costo 2

(R3, R5) con costo 2

(R5, R3) con costo 2

(R6, R5) con costo 2

(R5, R6) con costo 2

Avisos de estado de enlace de R3 para enviar a la red dorsal y a R1 y R2.

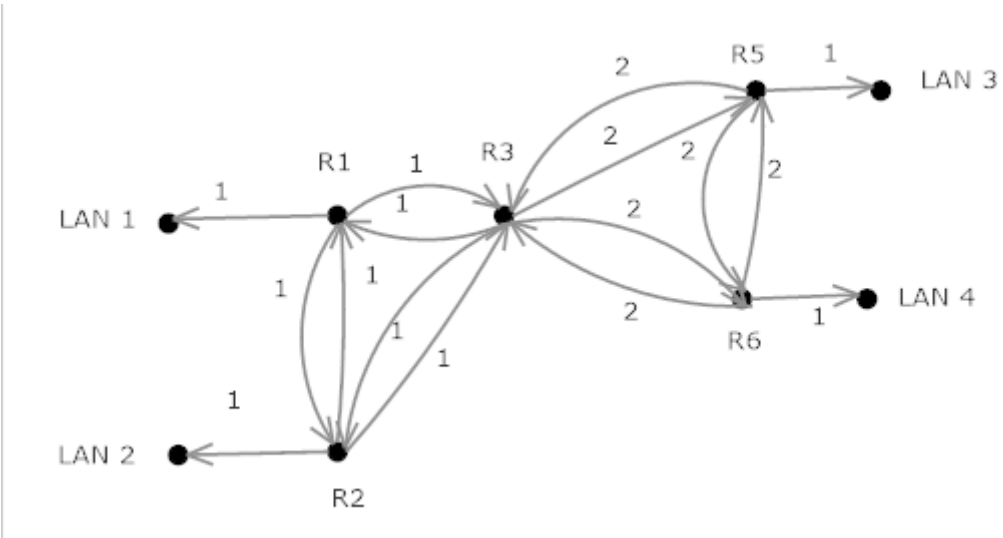
R3 manda resumen de área A1.

- A LAN 1: 2
- A LAN 2: 2

R3 manda paquete de estado de enlace a R1 y R2.

- A R1: costo 1
- A R2: costo 1
- R3 manda resúmenes de área A2, de área A3 y de área dorsal.

Grafo que calcula R2

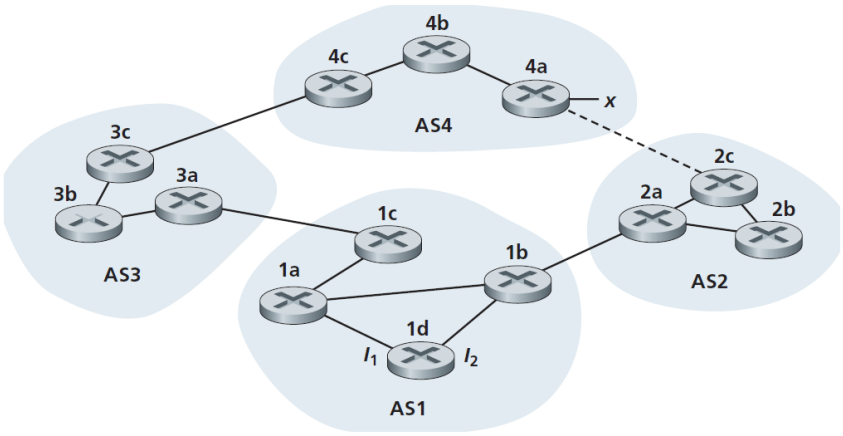


Grafo que calcula R3

Solo cambia parte del área dorsal donde se pone topología entera de ella con R4 incluido.

***Ejercicio H*:** Considerar la red que se muestra abajo. Suponer que AS3 y AS2 ejecutan OSPF como su protocolo de enrutamiento intra-SA. Suponer que AS1 y AS4 ejecutan RIP (protocolo parecido al enrutamiento de vector de distancia) como su protocolo de enrutamiento intra-SA donde cada enlace tiene costo 1. Suponer que eBGP e iBGP son usados para el protocolo de enrutamiento inter-SA. Una vez que el enrutador 1d aprende acerca de *x* va a poner una entrada (*x*, *I*) en su tabla de reenvío.

a. ¿Va a esa entrada tener *I* igual a *I*1 o a *I*2? Explicar el por qué en una oración.
b. Ahora suponer que hay un enlace físico entre AS2 y AS4, mostrado en la línea de guiones. Suponga que el enrutador 1d aprende que *x* es accesible vía AS2 y vía AS3. ¿Va a ser *I* ser fijad a *I*1 o a *I*2? Explicar el por qué en una oración.
c. Ahora suponga que hay otro SA AS llamado AS5, que yace en el camino entre AS2 y AS4 (no mostrado en el diagrama). Suponga que el enrutador 1d aprende que *x* es accesible vía AS2 AS5 AS4 así como AS3 AS4. ¿Va a ser *I* fijado a *I*1 o a *I*2? Explicar el por qué en una oración.



P38. Referring to the previous problem, once router 1d learns about *x* it will put an entry (*x*, *I*) in its forwarding table.

a. El único camino a x es: 3a, AS3, AS4, x. El camino más corto a 1c es por I1; luego I = I1 – usamos peso de los arcos 1.
b. Hay dos caminos a x: 3a, AS3, AS4, x y 2a , AS2, AS4, x. Como tienen la misma cantidad de SAs hay que usar hot potato routing para elegir el mejor camino: como 1b está mas cerca de 1d que 1c se elige el segundo camino. Para salir hacia 2a la ruta más corta sale por I2. Luego I =I2.
c. Hay dos caminos a x: 3a, AS3, AS4, x y 2a , AS2, AS5, AS4, x. Como el primero pasa por menos SAs, entonces se elige el primero. Luego el argumento es como en el item a. I=I1.