

Ruteo “Routing”

Contenido

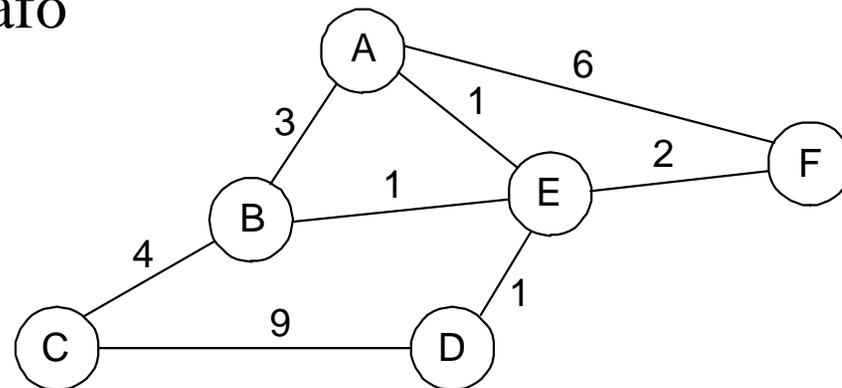
Algoritmos

Escalabilidad

Generalidades

- Re-envío versus Ruteo
 - Re-envío: debe seleccionar una puerta de salida basado en la dirección destino y las tablas de ruteo.
 - ruteo: proceso mediante el cual las tablas de ruteo son construidas.

- Vemos la red como un Grafo



- Problema: Encontrar el camino de menor costo entre dos nodos
- Factores
 - estáticos: topología
 - dinámicos: carga de los nodos y enlaces

Vector de Distancia

(Ejemplo:RIP, Routing Information Protocol)

- Cada nodo mantiene un conjunto de tríos
 - (Destination, Cost, NextHop)
- Intercambia actualizaciones con sus vecinos directamente conectados
 - Periódicamente (en el orden de varios segundos)
 - Cuando su tabla cambia (se habla de una actualización gatillada)
- Cada actualización es una lista de pares:
 - (Destination, Cost)
- Se actualiza tabla local si se recibe una mejor ruta
 - Costo menor
 - llegó desde el host próximo “next-hop”
- Se refrescan rutas existentes; se borran si hay time out

Ejemplo

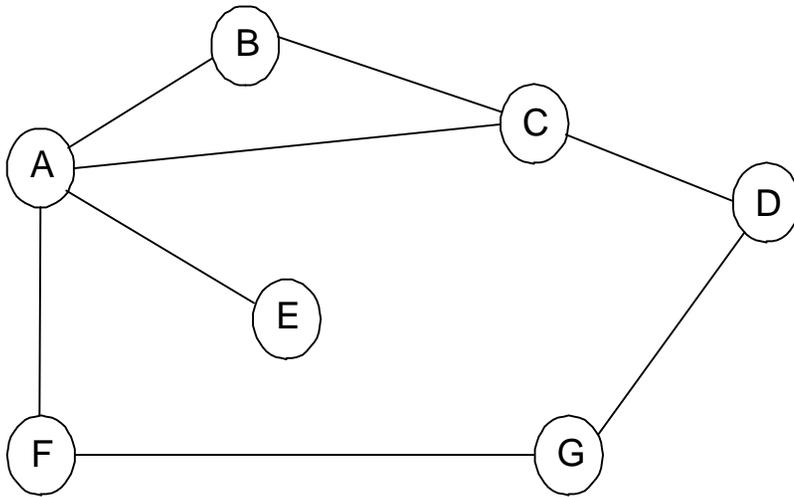
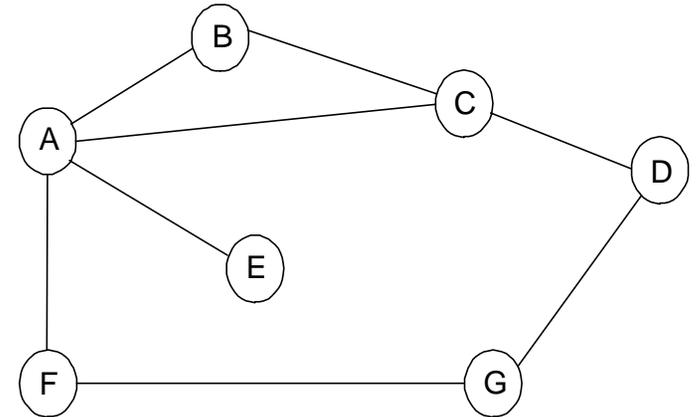


Tabla de ruta para B

Destination	Cost	NextHop
A	1	A
C	1	C
D	2	C
E	2	A
F	2	A
G	3	A

Ciclos de actualización



- Ejemplo 1
 - F detecta que enlace a G ha fallado
 - F fija distancia a G infinita y envía una actualización a A
 - A fija distancia a G infinita porque A usa F para llegar a G
 - A recibe actualización periódica de C con camino de 2 hops a G
 - A fija distancia a G como 3 y envía actualización a F
 - F decide él puede llegar a G en 4 hops vía A
- Ejemplo 2
 - Enlace de A a E falla.
 - A comunica distancia infinita a E y concurrentemente ..
 - B y C comunican distancia 2 a E
 - B decide que puede llegar a E en 3 hops vía C; comunica esto a A
 - A decide que puede llegar a E en 4 hops; comunica esto a C
 - C decide que puede llegar a E en 5 hops...

Heurísticas para romper los ciclos

- Fijar 16 como infinito, es decir cuando el costo llega a 16 se asume que no hay ruta al nodo
- Partir el horizonte (Split horizon): omite la información de distancia que fue obtenida del nodo al cual se le envía el vector.
- Partir el horizonte con reverso venenoso (Split horizon with poison reverse): incluye las entradas obtenidas desde el nodo al cual se envía el vector pero esos destinos se les pone costo infinito.
- Las últimas dos técnicas sólo operan cuando el lazo involucra dos nodos.
- La convergencia de este protocolo no es buena, se mejora con ruteo usando el estado de los enlaces.

Estado del Enlace

Ejemplo: OSPF, Open Shortest Path First Protocol

- Estrategia
 - Enviar a todos los nodos (no sólo los vecinos) información sobre enlaces directamente conectados (no la tabla completa). “se *inunda*”
- Paquete del estado del enlace (Link State Packet, LSP)
 - id del node que creó el LSP
 - costo del enlace a cada vecino directamente conectado
 - número de secuencia (SEQNO)
 - time-to-live (TTL) para este paquete

Estado del Enlace (cont)

- Inundación Confiable
 - almacena el LSP más reciente de cada nodo
 - re-envía LSP a todos excepto a quien me lo envió
 - genera un nuevo LSP periódicamente
 - incrementa SEQNO
 - inicia SEQNO en 0 cuando reboot
 - decrementa TTL de cada LSP almacenado
 - descarta cuando TTL=0

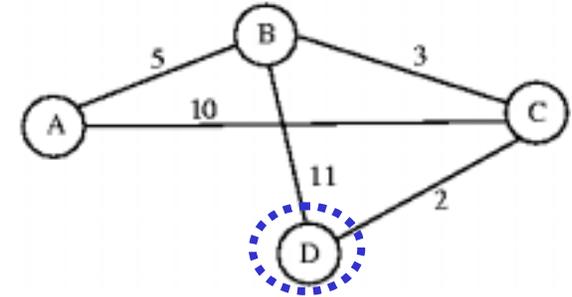
Cálculos de Ruta

- Algoritmo de Dijkstra para el camino más corto entre nodos
- Sea
 - N denota el conjunto de nodos del grafo
 - $l(i, j)$ denota costo no negativo (peso) para arco (i, j) . Si no hay arco, el costo es infinito.
 - s denota este nodo
 - M denota el conjunto de nodos incorporados hasta ahora
 - $C(n)$ denota el costo del camino de s a n

```
 $M = \{s\}$   
for each  $n$  in  $N - \{s\}$   
     $C(n) = l(s, n)$   
while ( $N \neq M$ )  
     $M = M$  unión  $\{w\}$  tal que  $C(w)$  es el mínimo para todo  $w$   
    en  $(N - M)$   
    for each  $n$  in  $(N - M)$   
         $C(n) = \text{MIN}(C(n), C(w) + l(w, n))$ 
```

Uso del algoritmo de Dijkstra en ruteo

- En la práctica el cálculo de la tabla de ruta se hace conforme llegan los LSP (link-state packet).
- Se manejan dos listas: la de entradas *tentativas* y las *confirmadas*. Ejemplo:



Paso	Confirmado	Tentativo	Comentario
1	(D,0,-)		Miramos el LSP de D porque es el único nuevo miembro confirmado
2	(D,0,-)	(B,11,B) (C,2,C)	El LSP de D nos dice cómo llegamos a B y C
3	(D,0,-) (C,2,C)	(B,11,B)	El miembro de menor costo entre los tentativos es C. Se examina el LSP del nodo recién confirmado
4	(D,0,-) (C,2,C)	(B,5,C) (A,12,C)	Con C, se actualizan los costos. Ahora llegamos a B vía C y se incorpora A
5	(D,0,-) (C,2,C) (B,5,C)	(A,12,C)	Se mueve el de menor costo de tentativos a confirmados.
6	(D,0,-) (C,2,C) (B,5,C)	(A,10,C)	Con B, se actualiza los costos. Ahora se llega a A vía B. Después de esto se mueve el menor costo de los tentativos a los confirmados. A es el único y terminamos.

Comparación entre algoritmo vector de distancia y estado de enlaces.

- En el algoritmo vector de distancia cada nodo transmite a sus vecinos lo que sabe respecto de toda la red (distancia a todos los nodos).
- En el algoritmo estado de enlaces cada nodo transmite a toda la red lo que sabe de sus vecinos. (el estado de sus vecinos)
- El segundo es **estable**, no genera gran tráfico y responde rápido a cambios de topología.
- El problema del segundo es la cantidad de información almacenada en los nodos.(un LSP por nodo)

Métricas

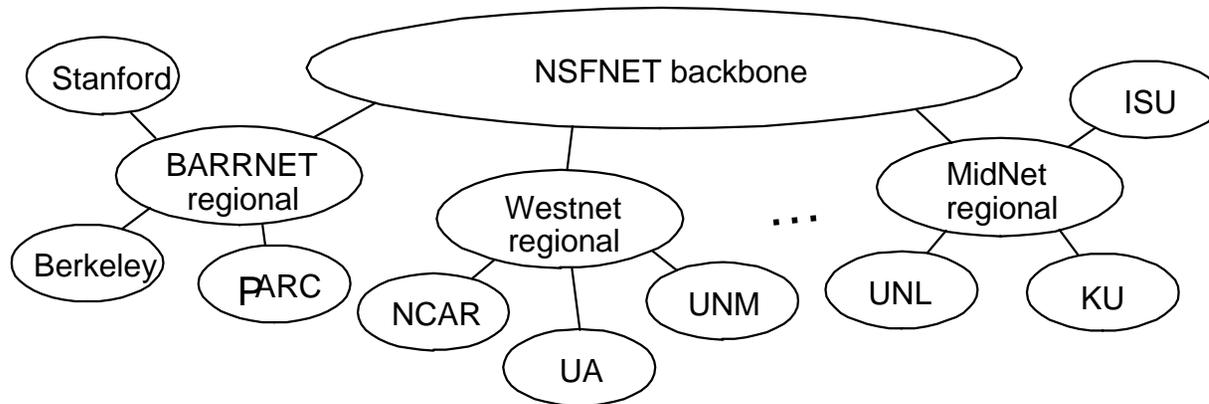
- Es forma como definimos el costo del enlace.
- Por ejemplo asignar 1 a cada enlace.
- Métrica ARPANET original
 - mide el número de paquetes encolados en cada enlace
 - No toma en cuenta latencia ni ancho de banda
- Métrica ARPANET nueva
 - Marca cada paquete entrante con su tiempo de llegada (arrival time, **AT**)
 - Graba tiempo de salida (departure time, **DT**)
 - Cuando llega el ACK del enlace de datos llega, calcula
$$\text{Delay} = (\text{DT} - \text{AT}) + \text{Transmit} + \text{Latency}$$
Transmit y Latency son parámetros estáticos del enlace.
 - Si hay timeout, reset **DT** a tiempo de salida para retransmisión
 - Costo del enlace = retardo promedio medido en algún período
- Mejora fina
 - Reducir el rango dinámico para el costo (se impone una saturación para el costo)
 - En lugar de retardo se emplea la utilización del enlace

Cómo hacer ruteo Escalable

- Direccionamiento plano versus jerárquico
- Ineficiente uso del espacio de direcciones jerárquicas
 - clase C con 2 hosts ($2/255 = 0.78\%$ eficiencia)
 - clase B con 256 hosts ($256/65535 = 0.39\%$ eficiencia)
- Aún hay demasiadas redes
 - Las tablas de ruteo no escalan
 - protocolos de propagación de rutas no escalan

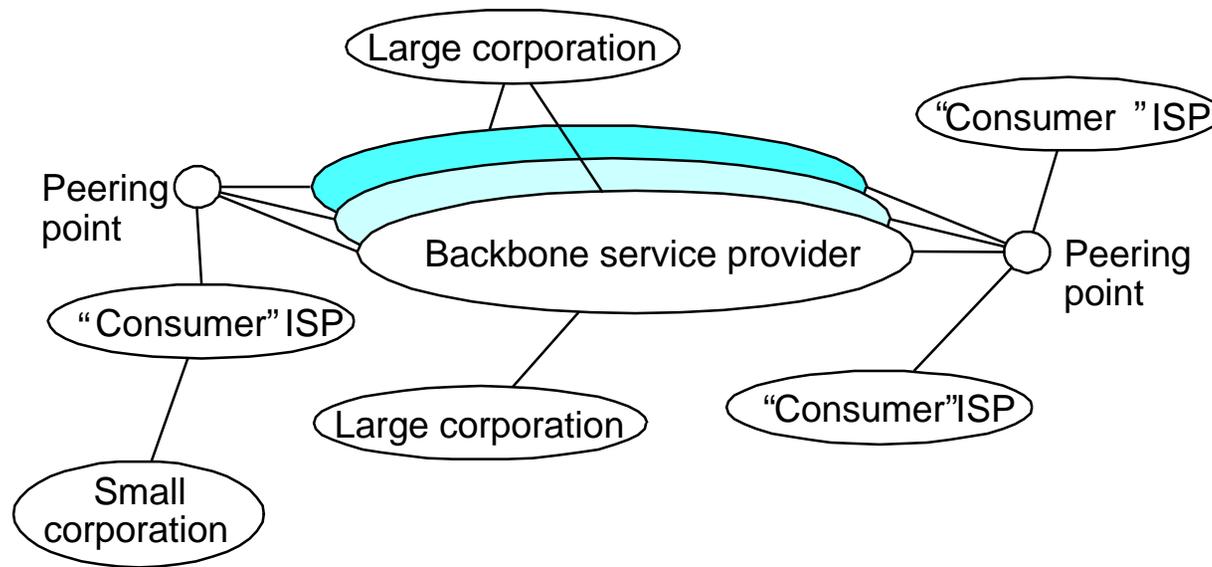
Estructura Internet

Pasado Reciente (1990, EEUU)



Estructura Internet

Hoy



Subredes

- Surgen como una necesidad de poder administrar mejor las direcciones IP asignadas dentro de una clase.
- Agrega otro nivel a la jerarquía dirección/ruta: *subred*
- *Máscara de Subred* define partición variable en la parte host
- Subredes son visibles sólo dentro del sitio, dentro de una misma red.

Network number	Host number
----------------	-------------

Class B address

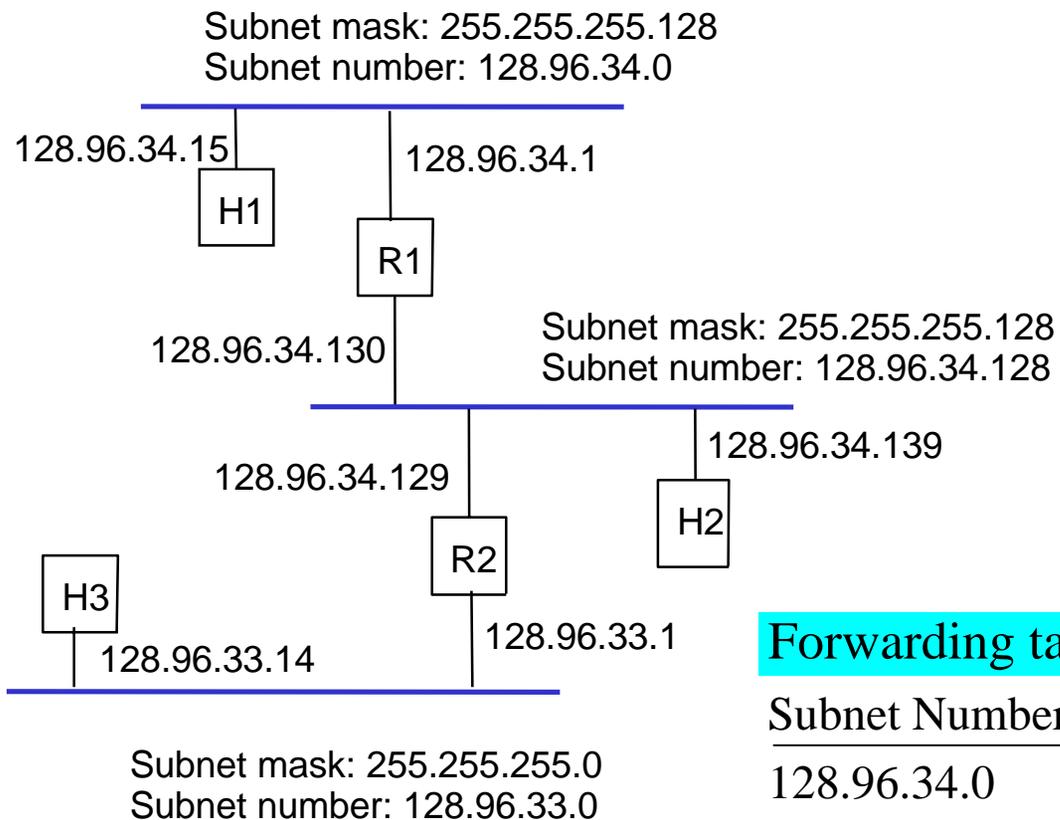
11111111111111111111111111111111	00000000
----------------------------------	----------

Subnet mask (255.255.255.0)

Network number	Subnet ID	Host ID
----------------	-----------	---------

Dirección de Subred

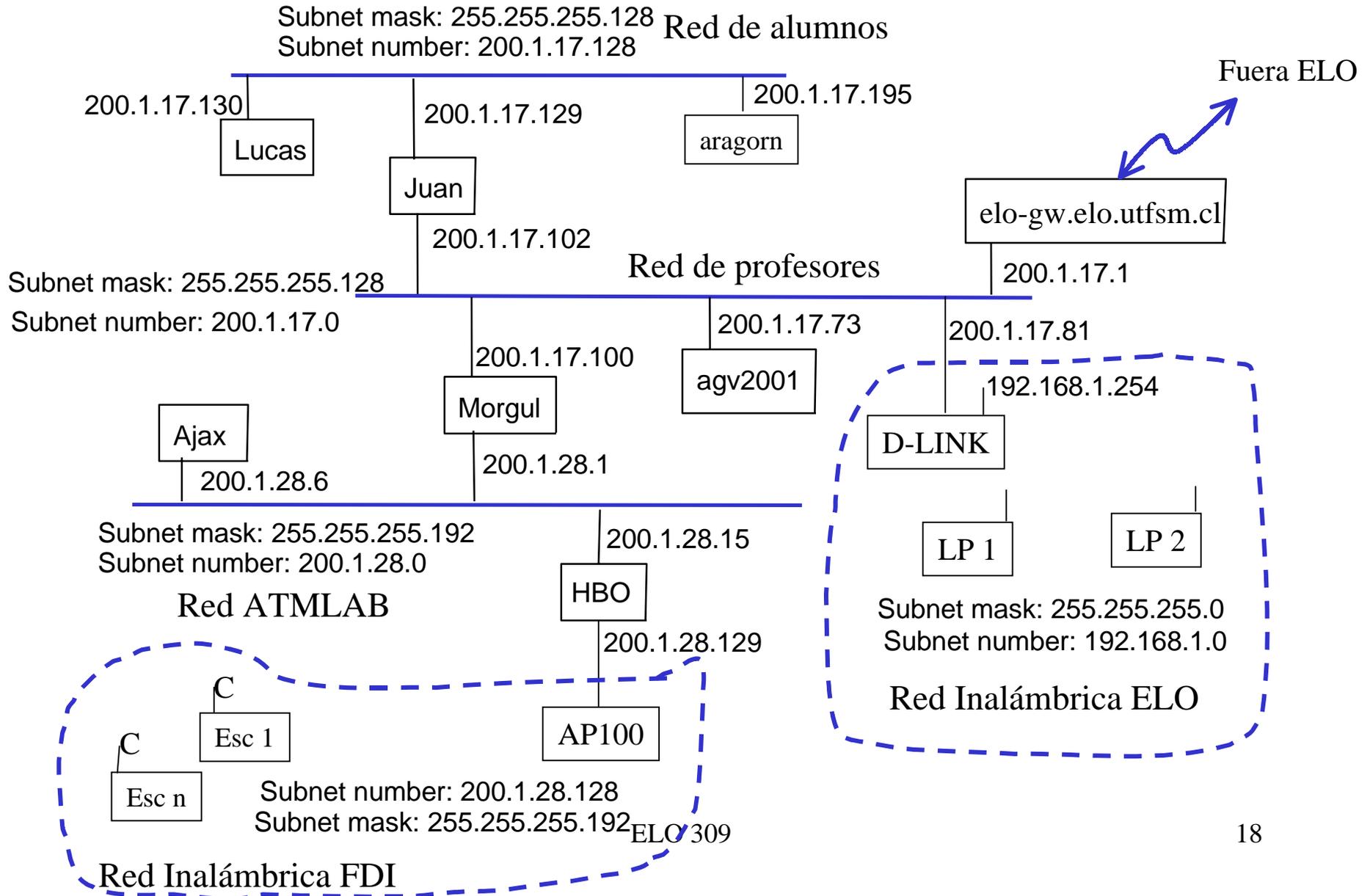
Ejemplo Subred



Forwarding table at router R1

Subnet Number	Subnet Mask	Next Hop
128.96.34.0	255.255.255.128	interface 0
128.96.34.128	255.255.255.128	interface 1
128.96.33.0	255.255.255.0	R2

Ejemplo Subredes en ELO



Algoritmo de Re-envío

```
D = Dirección IP destino
for each entrada (SubnetNum, SubnetMask, NextHop)
  D1 = SubnetMask & D
  if D1 = SubnetNum
    if NextHop is una interface
      Enviar datagrama directamente a to D
    else
      enviar datagrama a NextHop
```

- Se usa un router por omisión si nada coincide
- No necesariamente todos los 1s en la máscara de subred son contiguos
- Se puede poner múltiples subredes en una red física
- Subredes no son visibles para el resto de la Internet

Superredes

- Se asignan bloques de números de redes contiguos a redes cercanas.
- Surgen como una necesidad de reducir el número de entradas en las tablas de ruteo.
- Llamado CIDR: Classless Inter-Domain Routing
- Se representa un bloque con un par único
(**first_network_address, count**)
- Restringe bloques a potencias de 2
- Use una máscara de bits (CIDR mask) para identificar el tamaño del bloque
- Todos los routers deben entender direccionamiento CIDR
- Si en la tabla hay varias entradas que parecen conducir a un destino, tomar aquella con coincidencia mayor en la dirección de red.

Propagación de rutas

- Hay que conocer un router más astuto
 - hosts conocen el router local
 - el router local conoce los routers del sitio
 - los routers del sitio conocen los router centrales
 - Los routers centrales conocen todo
- Sistemas Autonomos (Autonomous System,AS)
 - Corresponde a un domino administrativo
 - Ejemplos: Universidad, compañía, backbone network
 - Se asigna a cada AS un número de 16-bits
- Jerarquía de dos niveles de propagaciones de rutas
 - Protocolo de rutas interiores (cada AS selecciona el propio)
 - Protocolo de rutas exterior (estándar en Internet)

Protocolos Populares de Gateway Interior

- RIP: Route Information Protocol
 - Distribuido con Unix
 - usa algoritmo vector distancia
 - basado en cuenta de hops
- OSPF: Open Shortest Path First
 - reciente estándar en Internet
 - usa el algoritmo de estado de enlaces
 - soporta balance de carga
 - soporta autenticación

Protocolo de Ruteo entre Dominios o Sistemas Autónomos

EGP: Exterior Gateway Protocol

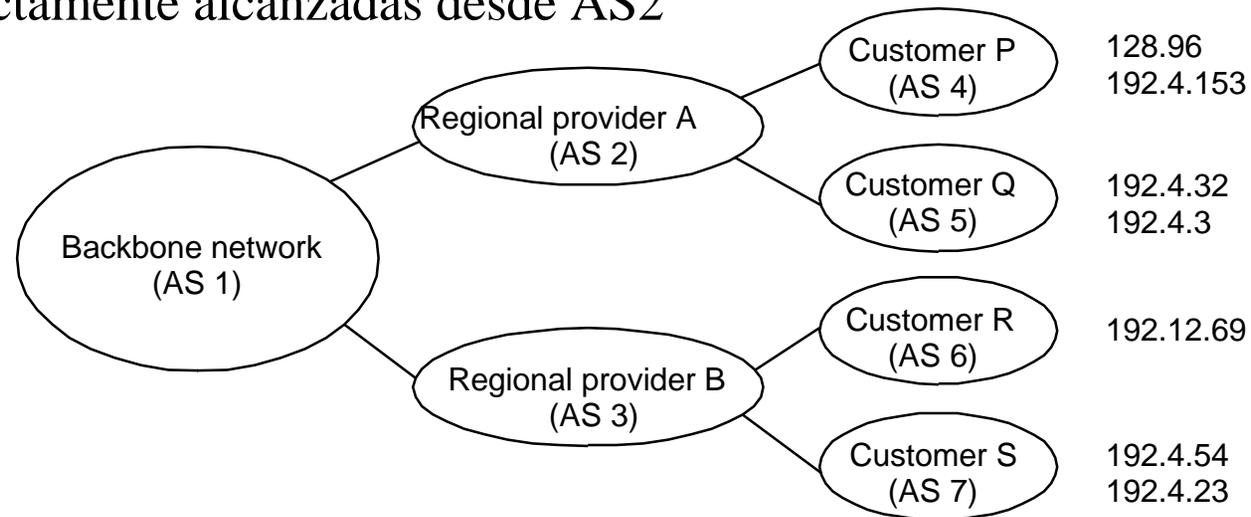
- Generalidades
 - diseñado para una Internet estructurada como árbol
 - se preocupa de *alcanzar* los nodos, no optimiza rutas
- Mensajes del Protocolo
 - Adquisición de vecinos: un router requiere que otro sea su par; pares intercambian información de alcance
 - Alcance de vecinos: un router periódicamente prueba si el otro es aún alcanzable; intercambia mensajes HELLO/ACK;
 - actualización de rutas: pares periódicamente intercambian sus tablas de ruteo (vector distancia)

BGP-4: Border Gateway Protocol

- Tipos AS (sistemas autónomos)
 - stub AS: tiene una única conexión a otro AS
 - transporta sólo tráfico local
 - multihomed AS: tiene conexiones a más de un AS
 - no transporta tráfico en transito
 - transit AS: tiene conexiones a más de un AS
 - transporta ambos tráfico local y en transito
- Cada AS tiene:
 - Uno o más routers de borde
 - Un portavoz BGP que publica:
 - Redes locales
 - Otras redes alcanzables (sólo el transit AS)
 - entrega información de rutas (*path*)

Ejemplo: BGP

- Portavoz de AS2 publica alcanzabilidad a P y Q
 - redes 128.96, 192.4.153, 192.4.32, y 192.4.3, pueden ser directamente alcanzadas desde AS2

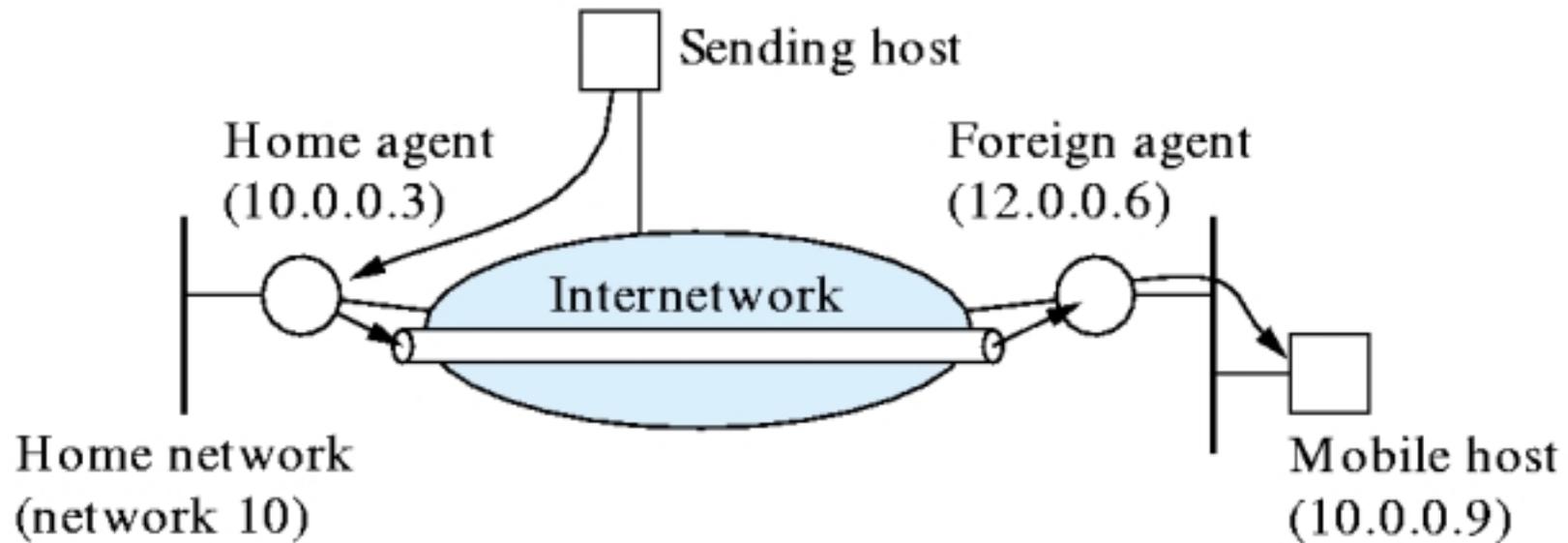


- Portavoz para backbone publica
 - redes 128.96, 192.4.153, 192.4.32, y 192.4.3 pueden ser alcanzadas a lo largo de la ruta (AS1, AS2).
- Portavoz puede cancelar una ruta publicada previamente

Soporte de Movilidad de Hosts

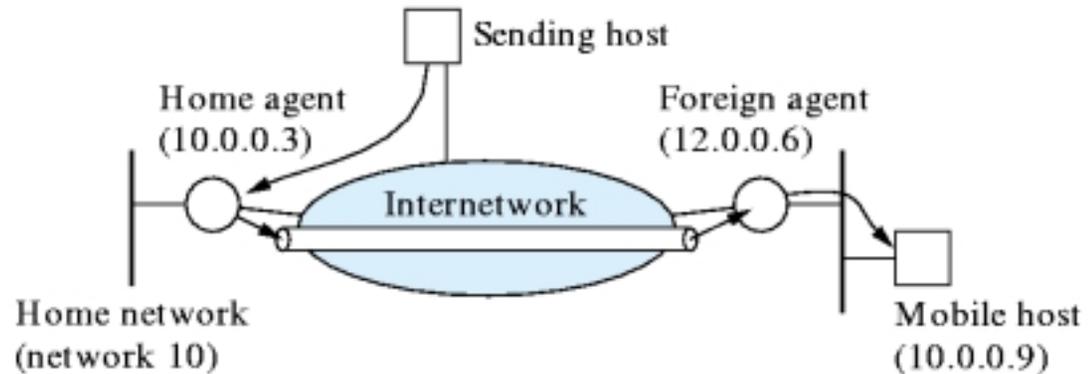
- La dirección de una máquina posee su dirección de red y máquina dentro de esa red.
- La dirección de red nos dice a qué red él está conectado.
- ¿Qué pasa si cambiamos una máquina de una red a otra?
- Si no cambiamos su IP, el host no es alcanzable!!
- ¿Como conseguimos que las aplicaciones de ese host sigan corriendo cuando nos cambiamos a otra red?
- El IETF ha propuesto una arquitectura basa en:
 - Que la solución no requiera cambios en las aplicaciones.
 - Que no se requiera cambiar la mayoría de los router de la red.
- Se requiere al menos cambiar un router, uno perteneciente a la red de la IP de la máquina que se moviliza.
- La idea es que la IP del móvil no cambie.
- La arquitectura es:

Soporte de Movilidad de Hosts



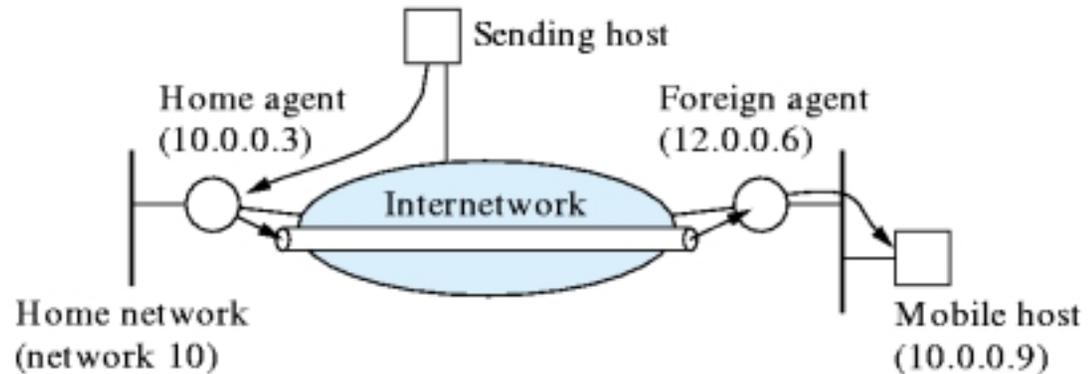
- Home agent: Router de la red a la cual pertenece la IP del móvil.
- Foreign agent: Router o máquina de la red remota.
- Home y Foreign agents anuncian periódicamente su presencia.
 - Anuncio del home permite al móvil conocerlo antes de marcharse
 - Anuncio del foreign permite al móvil conocerlo cuando llega a otra red.
- Cuando el móvil llega a la red remota, le informa al foreign sobre su home agent.
- El foreign le informa al home la presencia del móvil.

Soporte de Movilidad de Hosts



- Cuando el home agent sabe del móvil en otra red, personaliza al móvil usando la técnica “Proxy ARP”. En ésta él hace creer al resto que la IP del móvil posee la MAC del Home agent.
- Entonces, cuando un host fijo envía un datagrama al móvil, éste llega al home agent.
- El home agent lo encapsula en otro datagrama y lo envía al foreign agent.
- El foreign agent lo extrae y reconoce la dirección IP del móvil previamente registrado, entonces se lo envía a su dirección MAC directamente.

Soporte de Movilidad de Hosts



- El móvil responde directamente al host fijo sin y usando su IP como fuente.
- Depuración:
 - El móvil podría correr también el Foreign agent.
 - En este caso el móvil obtendría una IP dinámica usando DHCP sólo para recibir los datagramas desde el home agent, pero sería visto hacia el exterior como una máquina con su IP fija de origen.
- Los dos extremos podrían ser móviles y el mecanismo anterior se aplica para ambos.

IP Versión 6

- Características
 - Direcciones de 128-bit (Sin clases)
 - multicast
 - Servicio de tiempo real
 - autenticación y seguridad
 - autoconfiguración
 - Fragmentación de extremo a extremo
 - Permite extensiones del protocolo
- Encabezado
 - 40-byte encabezado base
 - Encabezados de extensión (orden fijo, largo generalmente fijo)
 - fragmentación
 - ruteo de fuente
 - autenticación y seguridad
 - otras opciones