

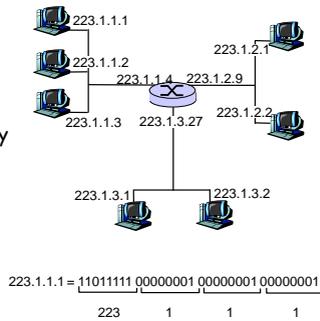
## Cap. 4: Capa de red

- 4.1 Introducción
- 4.2 circuitos virtuales y datagramas
- 4.3 dentro de un router
- 4.4 IP: Internet Protocol
  - formato de datagramas
  - direccionamiento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de enrutamiento
  - Link state
  - Distance Vector
  - Enrutamiento jerárquico
- 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast y multicast

Int. Redes de Computadores-Capa de Red 4-37

### Direccionamiento IP: introducción

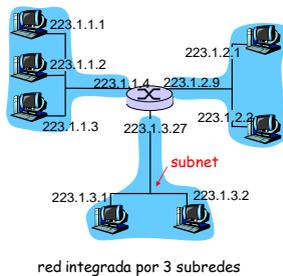
- dirección IP:
  - identificador de 32-bit para una *interfaz* de host o router
- *interfaz*: conexión entre el host/router y el enlace físico
  - un router tiene típicamente muchas interfaces
  - un host tiene típicamente una sola interfaz
  - una dirección IP asociada a cada interfaz



Int. Redes de Computadores-Capa de Red 4-38

### Subredes

- dirección IP:
  - subred (bits de mayor orden)
  - host (bits de menor orden)
- *qué es una subred?*
  - dispositivos cuya parte de subred de la dirección IP coincide...
  - ... pueden alcanzarse sin la intervención de un router (están en el "mismo cable", como una LAN hogareña)

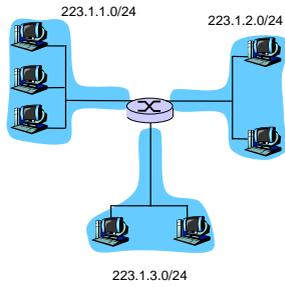


Int. Redes de Computadores-Capa de Red 4-39

## Subredes

### "Receta"

- Para determinar las subredes, desconectar cada interfaz de su host o router, creando islas de redes aisladas. Cada una de ellas en una **subred**.



Máscara de subred: /24

Int. Redes de Computadores-Capa de Red 4-40

---

---

---

---

---

---

---

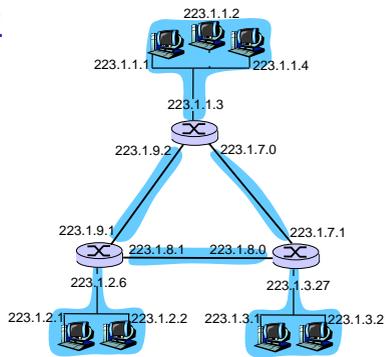
---

---

---

## Subredes

### Cuántas?



Int. Redes de Computadores-Capa de Red 4-41

---

---

---

---

---

---

---

---

---

---

## direccionamiento IP: CIDR

### CIDR: Classless InterDomain Routing

- porción de subred de la dirección de largo arbitrario
- formato de la dirección: a.b.c.d/x, donde x es el no. de bits en la parte de subred de la dirección



Int. Redes de Computadores-Capa de Red 4-42

---

---

---

---

---

---

---

---

---

---

## cómo obtener una dirección IP?

**P:** Cómo hace un host para obtener una dirección IP?

- "hard-coded" por el administrador de sistemas
  - Windows: "control-panel->network->configuration->tcp/ip->properties"
  - UNIX: /etc/rc.config o similar
- **DHCP:** Dynamic Host Configuration Protocol: obtención dinámica de una dirección, entregada por un servidor
  - "plug-and-play"

Int. Redes de Computadores-Capa de Red 4-43

---

---

---

---

---

---

---

---

## DHCP: Dynamic Host Configuration Protocol

**objetivo:** permite a un host obtener una dirección IP *dinámicamente* de un servidor cuando se une a la red

Renueva "lease" si la dirección está en uso

Permite reuso de direcciones (solo se mantiene una dirección mientras el host está conectado y "encendido")

Soporte de usuarios móviles cuando "llegan" a una red

Características del DHCP:

- host hace broadcast de mensaje "DHCP discover"
- servidor DHCP responde con mensaje "DHCP offer"
- host pide una dirección IP con el mensaje "DHCP request"
- servidor DHCP envía dirección en mensaje "DHCP ack"

Int. Redes de Computadores-Capa de Red 4-44

---

---

---

---

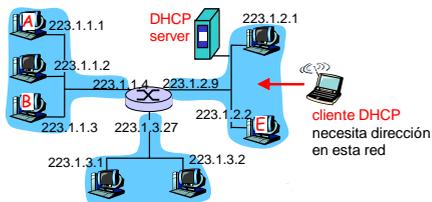
---

---

---

---

## Escenario DHCP cliente-servidor



Int. Redes de Computadores-Capa de Red 4-45

---

---

---

---

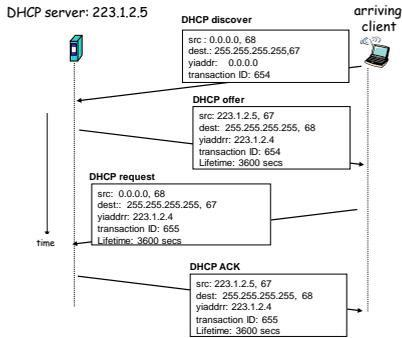
---

---

---

---

## Escenario DHCP cliente-servidor



Int. Redes de Computadores-Capa de Red 4-46

---

---

---

---

---

---

---

---

---

---

## cómo obtener una dirección IP?

- P:** como hace la red para obtener la parte de subred de la dirección IP?
- R:** recibe una porción del espacio de direccionamiento de su proveedor (ISP)

Bloque del ISP	11001000 00010111 00010000 00000000	200.23.16.0/20
Organización 0	11001000 00010111 00010000 00000000	200.23.16.0/23
Organización 1	11001000 00010111 00010010 00000000	200.23.18.0/23
Organización 2	11001000 00010111 00010100 00000000	200.23.20.0/23
...	.....	.....
Organización 7	11001000 00010111 00011110 00000000	200.23.30.0/23

Int. Redes de Computadores-Capa de Red 4-47

---

---

---

---

---

---

---

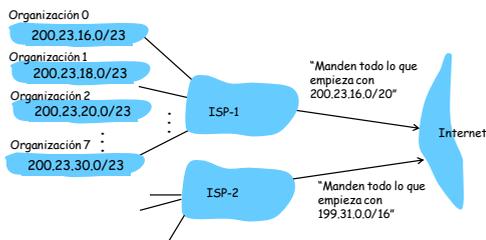
---

---

---

## Direccionamiento jerárquico: agregación de rutas

El direccionamiento jerárquico permite publicar en forma eficiente la información de enrutamiento:



Int. Redes de Computadores-Capa de Red 4-48

---

---

---

---

---

---

---

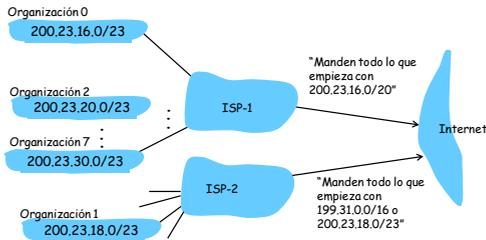
---

---

---

## Direccionamiento jerárquico: rutas más específicas

Organización 1 se mueve de ISP-1 a ISP-2



Int. Redes de Computadores-Capa de Red 4-49

---

---

---

---

---

---

---

---

---

---

## "gobierno" de la internet...

**P:** Cómo obtiene un ISP un bloque de direcciones?

**R:** ICANN: Internet Corporation for Assigned Names and Numbers

- asignar direcciones
  - gestionar DNS
  - asignar nombres de dominio, resolver disputas
- Regionalización
- AFRINIC, RIPE NCC, ARIN, APNIC, LACNIC

Int. Redes de Computadores-Capa de Red 4-50

---

---

---

---

---

---

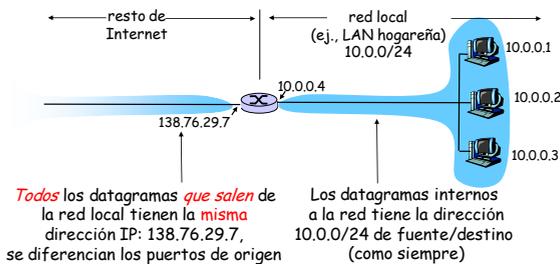
---

---

---

---

## NAT: Network Address Translation



Int. Redes de Computadores-Capa de Red 4-51

---

---

---

---

---

---

---

---

---

---

## NAT: Network Address Translation

- **Motivación:** la red local utiliza una sola dirección IP visto desde el mundo exterior:
  - no es necesario solicitar un rango de direcciones al ISP: solo una dir. IP para todos los dispositivos
  - se pueden cambiar direcciones de los dispositivos en la red local sin notificar al "resto del mundo"
  - Se puede cambiar de ISP sin modificaciones en la red local
  - los dispositivos en la red local no son "visibles" desde el mundo exterior (un extra de seguridad).

Int. Redes de Computadores-Capa de Red 4-52

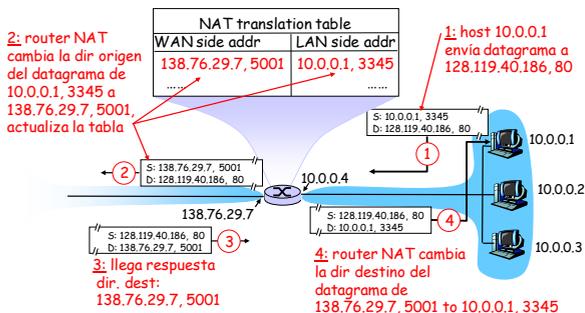
## NAT: Network Address Translation

**Implementación:** un router NAT debe:

- **datagramas salientes:** *reemplazar* (dir IP origen, port #) de cada datagrama a (dir IP del NAT, new port #) ... clientes/servidores remotos responderán usando (NAT IP address, new port #) como destino.
- **recordar (en la NAT translation table)** cada par de traslaciones (dir IP origen, port #), (dir IP del NAT, new port #)
- **datagramas entrantes:** *reemplazar* (dir IP NAT, new port #) en campos destino al correspondiente (dir IP origen, port #) almacenado en la tabla de NAT

Int. Redes de Computadores-Capa de Red 4-53

## NAT: Network Address Translation



Int. Redes de Computadores-Capa de Red 4-54

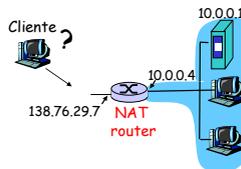
## NAT: Network Address Translation

- campo *port-number* de 16 bits:
  - 60,000 conexiones simultáneas con una sola dirección IP!
- NAT es contradictorio:
  - los routers solo deberían procesar hasta capa 3
  - viola la abstracción de extremo a extremo
    - el NAT debe ser tenido en cuenta por los diseñadores de aplicaciones, por ej. P2P
  - la carencia de direcciones debería resolverse por métodos más "limpios", como IPv6

Int. Redes de Computadores-Capa de Red 4-55

## como atravesar un NAT?

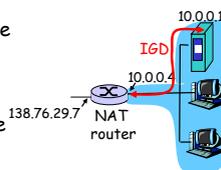
- un cliente se quiere conectar al servidor con dirección 10.0.0.1
  - pero esa dir. es local a la LAN, no se puede usar como dir. de destino
  - Solo se puede usar la dir del NAT: 138.76.29.7
- solución 1: configuración estática de *port forwarding*
  - ej., (123.76.29.7, port 2500) siempre se traduce a 10.0.0.1 port 25000



Int. Redes de Computadores-Capa de Red 4-56

## como atravesar un NAT?

- solución 2: Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Permite a un host detrás de un NAT a:
  - ❖ aprender la dirección IP pública (138.76.29.7)
  - ❖ agregar/remover mapeos de puertos (con tiempos de lease)



es decir, automatiza la configuración estática del port forwarding del NAT

Int. Redes de Computadores-Capa de Red 4-57

## como atravesar un NAT?

- solución 3: "relaying" (usado en Skype)
  - cliente "NATeado" establece conexión al relay
  - clientes externos se conectan al relay
  - relay hace "bridge" de paquetes entre conexiones



Int. Redes de Computadores-Capa de Red 4-58

---

---

---

---

---

---

---

---

---

---

## Cap. 4: Capa de red

- 4.1 Introducción
- 4.2 circuitos virtuales y datagramas
- 4.3 dentro de un router
- 4.4 IP: Internet Protocol
  - formato de datagramas
  - direccionamiento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de enrutamiento
  - Link state
  - Distance Vector
  - Enrutamiento jerárquico
- 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast y multicast

Int. Redes de Computadores-Capa de Red 4-59

---

---

---

---

---

---

---

---

---

---

## ICMP: Internet Control Message Protocol

- usado por hosts & routers para comunicar información e nivel de red
    - reporte de errores: *unreachable* host, red, puerto, protocolo
    - echo request/reply (usado por el ping)
  - capa de red "encima" de IP:
    - mensajes ICMP transportado en datagramas IP
  - mensajes ICMP: tipo, código, más los primeros 8 bytes del datagrama IP que causó el error
- | Type | Code | description                                   |
|------|------|---|
| 0    | 0    | echo reply (ping)                             |
| 3    | 0    | dest. network unreachable                     |
| 3    | 1    | dest host unreachable                         |
| 3    | 2    | dest protocol unreachable                     |
| 3    | 3    | dest port unreachable                         |
| 3    | 6    | dest network unknown                          |
| 3    | 7    | dest host unknown                             |
| 4    | 0    | source quench (congestion control - not used) |
| 8    | 0    | echo request (ping)                           |
| 9    | 0    | route advertisement                           |
| 10   | 0    | router discovery                              |
| 11   | 0    | TTL expired                                   |
| 12   | 0    | bad IP header                                 |

Int. Redes de Computadores-Capa de Red 4-60

---

---

---

---

---

---

---

---

---

---

## Traceroute e ICMP

- fuente envía una serie de segmentos UDP al destino
    - el 1o tiene TTL =1
    - el 2o tiene TTL=2, etc.
    - no. de puerto "raro"
  - cuando el n-simo datagrama arriba al n-simo router:
    - el router descarta el datagrama...
    - ...y envía a la fuente un mensaje ICMP (type 11, code 0)
    - el mensaje incluye el nombre y dir IP del router
  - cuando llega el mensaje ICMP, la fuente calcula el RTT
  - traceroute repite esta operación 3 veces
- criterio de parada**
- el segmento UDP eventualmente llega al host destino
  - este retorna el mensaje ICMP "host unreachable" (type 3, code 3)
  - cuando la fuente recibe este paquete ICMP, para.

Int. Redes de Computadores-Capa de Red 4-61

---

---

---

---

---

---

---

---

---

---

## Cap. 4: Capa de red

- 4.1 Introducción
- 4.2 circuitos virtuales y datagramas
- 4.3 dentro de un router
- **4.4 IP: Internet Protocol**
  - formato de datagramas
  - direccionamiento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de enrutamiento
  - Link state
  - Distance Vector
  - Enrutamiento jerárquico
- 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast y multicast

Int. Redes de Computadores-Capa de Red 4-62

---

---

---

---

---

---

---

---

---

---

## IPv6

- **motivación inicial:** el espacio de direcciones de 32 bits "se está por agotar."
  - **motivación adicional:**
    - formato del cabezal ayuda a acelerar el procesamiento/forwarding del paquete
    - cambios en el cabezal facilitan QoS
- formato del datagrama IPv6:**
- cabezal de largo fijo: 40 bytes
  - no se permite fragmentación

Int. Redes de Computadores-Capa de Red 4-63

---

---

---

---

---

---

---

---

---

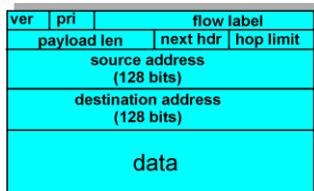
---

## Cabezal IPv6

**Priority:** identifica prioridad entre datagramas en un flujo

**Flow Label:** identifica datagramas en el mismo flujo (concepto de "flujo"...).

**Next header:** identifica el protocolo de capa superior



← 32 bits →

Int. Redes de Computadores-Capa de Red 4-64

---

---

---

---

---

---

---

---

---

---

## Otros cambios con respecto a IPv4

- Checksum:** eliminado para reducir procesamiento en cada hop
- Options:** permitido, pero fuera del header, indicado por el campo "Next Header"
- ICMPv6:** nueva versión de ICMP
  - Tipos de mensajes adicionales, por ej. "Packet Too Big"
  - Funciones de gestión de grupos de multicast

Int. Redes de Computadores-Capa de Red 4-65

---

---

---

---

---

---

---

---

---

---

## Transición de IPv4 a IPv6

- No se puede hacer una actualización de todos los routers simultáneamente
  - no hay "día D"
  - Cómo puede operar una red con routers IPv4 e IPv6 mezclados?
- Tunneling:** IPv6 transportado como *payload* en datagramas IPv4

Int. Redes de Computadores-Capa de Red 4-66

---

---

---

---

---

---

---

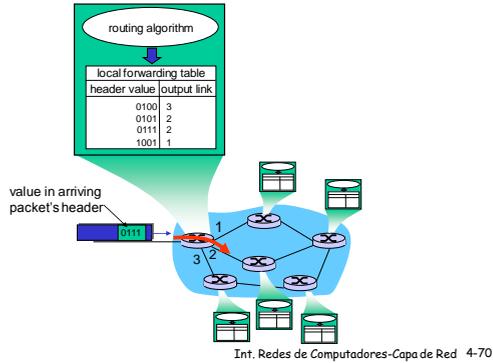
---

---

---



## Interacción entre routing & forwarding




---

---

---

---

---

---

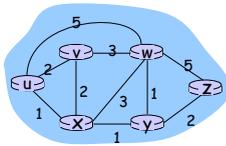
---

---

---

---

## Abstracción de grafo



Graph:  $G = (N, E)$

$N$  = conjunto de routers = { u, v, w, x, y, z }

$E$  = conjunto de enlaces = { (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

Nota: esta abstracción es útil en otros contextos de red

Ejemplo: P2P, donde  $N$  es el conjunto de pares y  $E$  el conjunto de conexiones TCP

Int. Redes de Computadores-Capa de Red 4-71

---

---

---

---

---

---

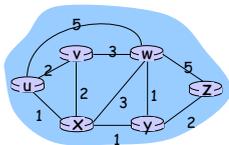
---

---

---

---

## Abstracción de grafo: costos



$c(x,x')$  = costo del enlace  $(x,x')$

- por ej.,  $c(w,z) = 5$

• algunas opciones para el costo:  
puede ser 1, con relación inversa al ancho de banda, con relación inversa a la congestión, entre otras

costo del camino  $(x_1, x_2, x_3, \dots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$

pregunta: cual es el camino con menor costo entre u y z ?

Algoritmo de enrutamiento:  
encuentra el camino de costo mínimo

Int. Redes de Computadores-Capa de Red 4-72

---

---

---

---

---

---

---

---

---

---

## Clasificación de algoritmos de enrutamiento

### Información global o descentralizada?

#### Global:

- todos los routers conocen la topología completa, y el costo de los enlaces
- algoritmos "link state"

#### Descentralizada:

- los router conocen los vecinos directamente conectados, y el costo de los enlaces a estos vecinos
- proceso de cómputo iterativo, con intercambio de información entre vecinos
- algoritmos "distance vector"

### Estático o dinámico?

#### Estático:

- las rutas cambian lentamente

#### Dinámico:

- Cambios más frecuentes en rutas
  - actualización periódica
  - en respuesta a cambios en topología o costo de los enlaces

Int. Redes de Computadores-Capa de Red 4-73

---

---

---

---

---

---

---

---

---

---

## Cap. 4: Capa de red

- 4.1 Introducción
- 4.2 circuitos virtuales y datagramas
- 4.3 dentro de un router
- 4.4 IP: Internet Protocol
  - formato de datagramas
  - direccionamiento IPv4
  - ICMP
  - IPv6

- 4.5 Algoritmos de enrutamiento
  - Link state
  - Distance Vector
  - Enrutamiento jerárquico
- 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast y multicast

Int. Redes de Computadores-Capa de Red 4-74

---

---

---

---

---

---

---

---

---

---

## Algoritmo de enrutamiento Link-State

### Algoritmo de

- Topología de la red y costos de los enlaces conocidos por todos los nodos
  - mediante "link state advertisements"
  - todos los nodos **Dijkstra** tienen la misma información
- se computan los caminos de costo mínimo entre un nodo (raíz) al resto de los nodos
  - determina la **tabla de forwarding** para ese nodo
- iterativo: luego de  $k$  iteraciones, se conocen los caminos de costo mínimo a  $k$  destinos

### Notación:

- $c(x,y)$ : costo del enlace entre nodos  $x,y$ ;  $= \infty$  si no son vecinos directos
- $D(v)$ : valor actual del costo del camino desde origen al destino  $v$
- $p(v)$ : nodo predecesor en el camino desde fuente a destino  $v$
- $N'$ : conjunto de nodos cuyo costo de camino se ha computado

Int. Redes de Computadores-Capa de Red 4-75

---

---

---

---

---

---

---

---

---

---

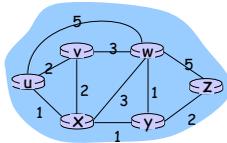
## Algoritmo de Dijkstra

- 1 **Initialization:**
- 2  $N' = \{u\}$
- 3 for all nodes  $v$
- 4 if  $v$  adjacent to  $u$
- 5 then  $D(v) = c(u,v)$
- 6 else  $D(v) = \infty$
- 7
- 8 **Loop**
- 9 find  $w$  not in  $N'$  such that  $D(w)$  is a minimum
- 10 add  $w$  to  $N'$
- 11 update  $D(v)$  for all  $v$  adjacent to  $w$  and not in  $N'$  :
- 12  $D(v) = \min(D(v), D(w) + c(w,v))$
- 13 /\* new cost to  $v$  is either old cost to  $v$  or known
- 14 shortest path cost to  $w$  plus cost from  $w$  to  $v$  \*/
- 15 **until all nodes in  $N'$**

Int. Redes de Computadores-Capa de Red 4-76

## Algoritmo de Dijkstra: ejemplo

Step	$N'$	$D(v),p(v)$	$D(w),p(w)$	$D(x),p(x)$	$D(y),p(y)$	$D(z),p(z)$
0	u	2,u	5,u	1,u	$\infty$	$\infty$
1	ux	2,u	4,x	2,x	$\infty$	$\infty$
2	uxy	2,u	3,y	4,y	4,y	$\infty$
3	uxyv	2,u	3,y	4,y	4,y	4,y
4	uxyvw	2,u	3,y	4,y	4,y	4,y
5	uxyvwz	2,u	3,y	4,y	4,y	4,y



Int. Redes de Computadores-Capa de Red 4-77

## Algoritmo de Dijkstra: ejemplo (cont)

"Shortest-path tree" resultante desde u:

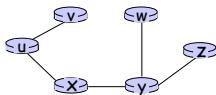


Tabla de forwarding resultante en u:

destino	enlace
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Int. Redes de Computadores-Capa de Red 4-78

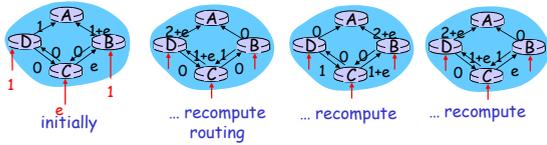
## Algoritmo de Dijkstra: discusión

**Complejidad del algoritmo:** n nodos

- cada iteración: necesita chequear todos los nodos, w, que no están en N
- $n(n+1)/2$  comparaciones:  $O(n^2)$
- implementación más eficiente posible:  $O(n \log n)$

**Posibles oscilaciones:**

- por ej., costo del enlace = cantidad de tráfico transportado



Int. Redes de Computadores-Capa de Red 4-79

---

---

---

---

---

---

---

---

---

---