Módulo 02
La Capa de Aplicaciones (Pt. 5)

Redes de Computadoras
Depto. de Cs. e Ing. de la Comp.
Universidad Nacional del Sur

Copyright

Contenidos

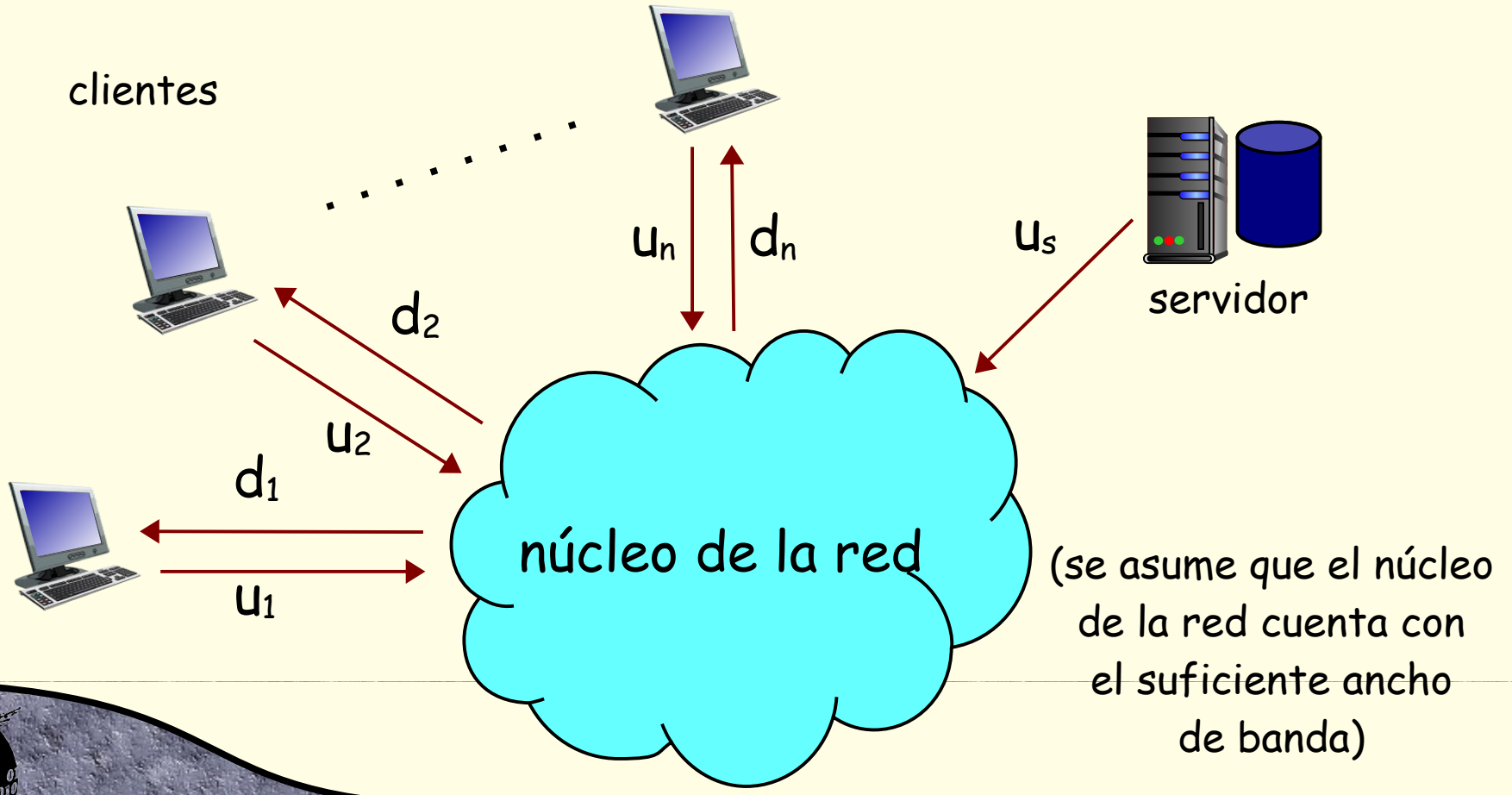
- Servicios que requiere la capa de aplicaciones
- Protocolos de la capa de aplicaciones
  - **HTTP**
  - **DNS**
  - **SMTP, POP e IMAP**
- Arquitectura de las aplicaciones **P2P**
- Programación basada en sockets

Arquitectura P2P pura

- Características de las aplicaciones que hacen uso del estilo arquitectónico **P2P** puro:
  - No requiere de servidores todo el tiempo en línea
  - Las computadoras en la frontera de la red se comunican directamente entre sí
  - Los pares se conectan y desconectan de la red **P2P** todo el tiempo, a veces modificando su dirección **IP**
- Analizaremos principalmente dos aspectos, la distribución de contenidos y la búsqueda de información en redes **P2P**

Distribución de contenidos

- ¿Cuánto tiempo toma distribuir un documento de **f** bits de un servidor a **n** computadoras?



Distribución de contenidos

- Análisis (modelo cliente servidor):
  - El servidor debe enviar secuencialmente **n** veces el documento de **f** bits a cada uno de los clientes
  - Esto es, insume alrededor de $\mathbf{nf/u_s}$ segundos
  - A un determinado cliente **i** le toma $\mathbf{f/d_i}$ segundos el descargar el documento
  - En síntesis, el tiempo que toma la distribución del documento bajo este modelo es:

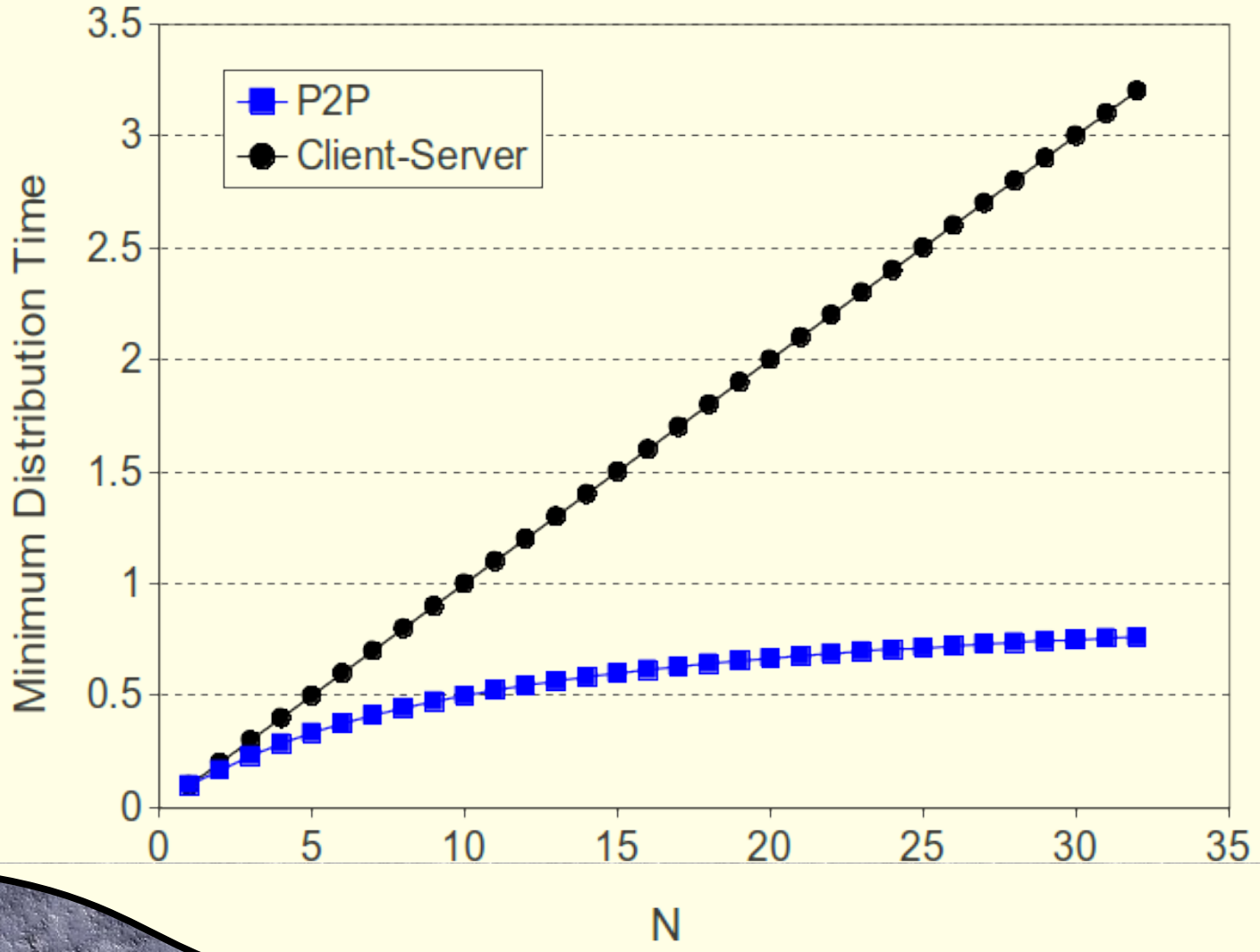
$$T_{c\text{-}s} = \text{máx}(nf/u_s, f/\text{mín}(d_i))$$

Distribución de contenidos

- Análisis (modelo **P2P**):
  - El servidor como mínimo debe enviar enviar al conjunto de clientes una copia entera de documento
  - Esto es, insume alrededor de f/u_s segundos
  - A un determinado cliente i le toma f/d_i segundos el descargar el documento
  - Los nf bits han de ser descargados de cualquier par, es decir, usando el ancho de banda agregado $u_s+\sum u_i$

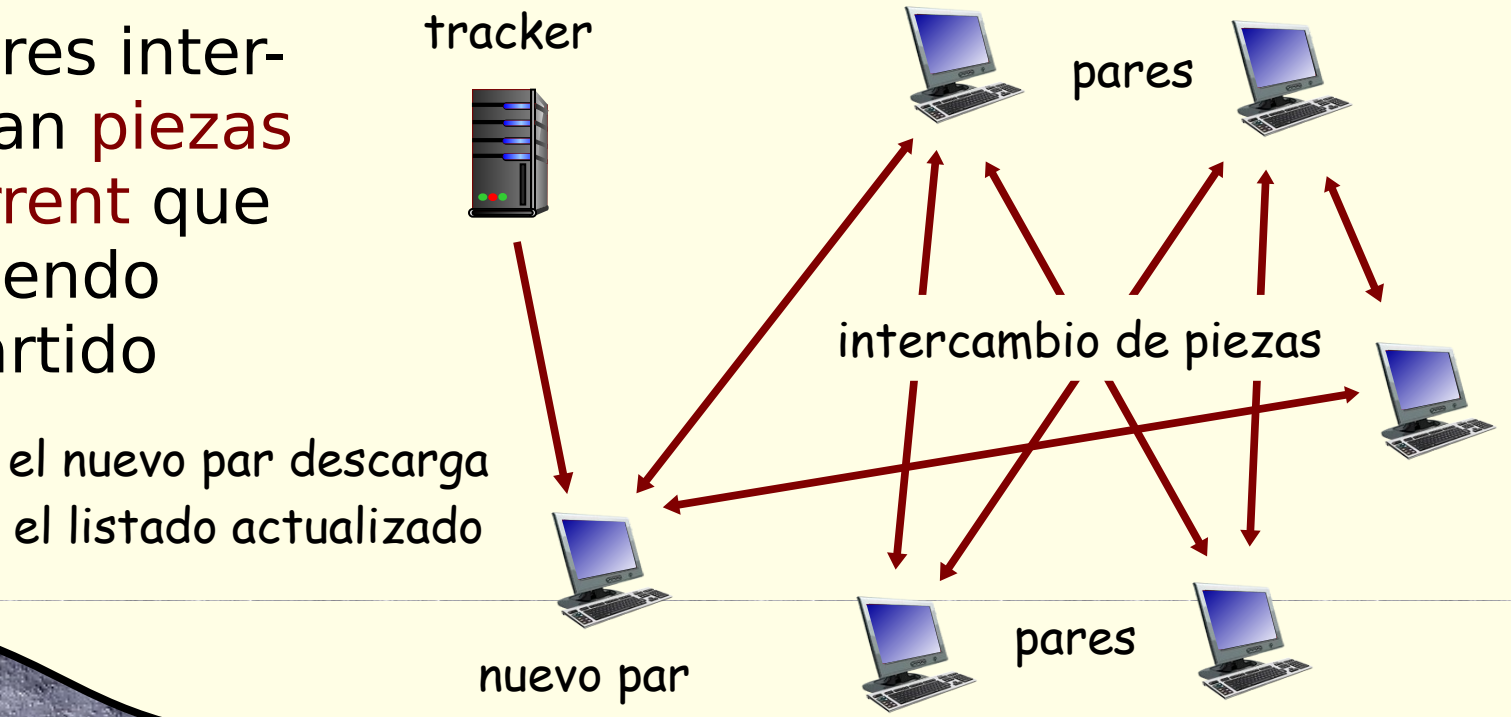
$$T_{p2p} = máx(f/u_s, f/mín(d_i), nf/(u_s+\sum u_i))$$

Distribución de contenidos



Protolo Bit Torrent

- El protocolo Bit Torrent ilustra el modelo **P2P** de distribución de contenido antes analizado
  - Un nodo destacado, denominado tracker, mantiene actualizado el listado de pares activos
  - Los pares inter-cambian piezas del torrent que está siendo compartido



Protocolo Bit Torrent

- Características del protocolo:
  - El torrent se divide en piezas de 256 KB
  - Al sumarse al torrent se debe registrar con el tracker, quien le acerca el listado de pares activos
  - Cada par se conecta con un subconjunto de los pares activos (denominados vecinos)
  - Al mismo tiempo que descarga nuevas piezas va compartiendo las que ya tiene con sus vecinos
  - Los pares pueden ingresar o salir del torrent a voluntad y en cualquier momento

Obtención de las piezas

- En todo momento, los distintos pares cuentan con distintos subconjuntos de piezas
- Periódicamente cada par consulta a sus vecinos acerca del listado de piezas que poseen
- Cada par al recibir esa información de sus vecinos procede a encargar le envíen las piezas que le estén faltando
  - Las piezas faltantes se ordenan en función de su rareza, encargando primero las menos difundidas
  - ¿Cuál sería la razón?

Envío de las piezas

- Para determinar que solicitudes priorizar, los pares ordenan a sus vecinos en función del ancho de banda con el cual éstos le estén enviando piezas (favor con favor se paga)
  - Cada 10 segundos se eligen los mejores cuatro vecinos y se atiende sus primeros pedidos
- En simultáneo, cada 30 segundos se elige otro par al azar y se le envía lo que necesite
  - La idea es agregarle dinamismo a la red de pares

Favor con favor se paga

- Supongamos que Andrea elige al azar a Bruno
  - Seguramente se convertirá en uno de los mejores vecinos de Bruno
  - De ser así, Bruno le empezará a subir a Andrea, y quizás llegue a ser uno de sus vecinos preferido

Base de datos simple

- Una base de datos en su versión más simple es en esencia un mapeo entre claves y valores:
  - Clave: apellido y nombre; Valor: número de teléfono

clave	valor
Palotes, Juan Dellos	154-354-3570
D'etal, Fulano	156-585-3791
Mengano, Evaristo	154-141-0902
…….	……
Zultano, Cástulo	155-341-0908

  - Clave: título de la película; Valor: dirección **IP**

Tabla hash

- Cualquier clave puede ser indexada como si se tratara de una clave numérica:
  - **clave numérica = hash(clave original)**

clave original	clave numérica	valor
Palotes, Juan Dellos	8962458	154-354-3570
D'etal, Fulano	7800356	156-585-3791
Mengano, Evaristo	1567109	154-141-0902
.....	.....	.....
Zultano, Cástulo	2360012	155-341-0908

Distributed Hash Table

- Un aspecto nada trivial en las aplicaciones **P2P** es la representación de bases de datos
  - En contraste, en una aplicación cliente-servidor, podemos hacer uso del primer **DBMS** que aparezca en una búsqueda en Google
- El mecanismo mayormente utilizado se denomina Distributed Hash Table (**DHT**)
  - La **DHT** hace las veces de base de datos distribuida
  - Se compone de un conjunto de tuplas (clave, valor) repartidas entre los pares

Distributed Hash Table

- Cualquier par puede consultar la base de datos mediante una clave
  - La **DHT** retornará el valor asociado a esa clave
  - Para resolver la consulta, se intercambia un pequeño número de mensajes entre los pares
- Cada par necesita conocer sólo un pequeño número de su pares
  - El algoritmo tolera que los pares entren y salgan de la red **P2P** en cualquier momento

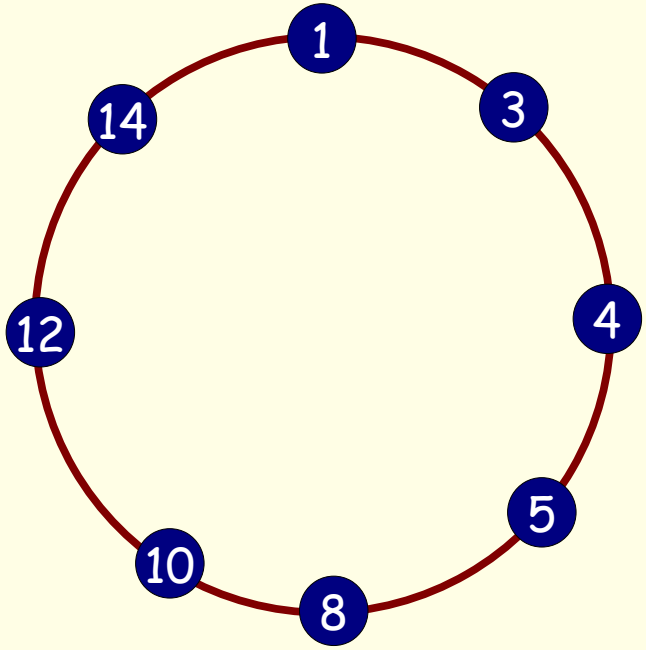
Identificadores DHT

- Los pares de la red **P2P** se identifican a través de un entero en el rango **[0, 2^n-1]**
  - Es decir, cada par se caracteriza mediante **n** bits
- A su vez, las claves de las tuplas tienen que ser valores enteros tomados del mismo rango
  - Como vimos, si las claves no fueran valores enteros, se puede simplemente calcular el valor de hash sobre la misma

Asignación de claves a pares

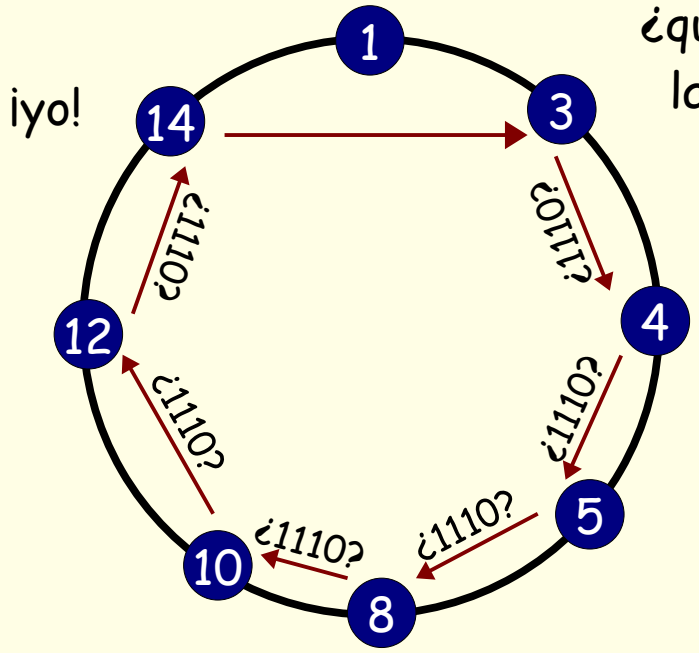
- El problema central al implementar una **DHT** es cómo repartir las tuplas entre los pares
  - Como pares y claves comparten el mismo rango, se puede ensayar asignar cada tupla al par cuyo identificador sea el más próximo a su clave
- En los siguientes ejemplos consideraremos como par más próximo al sucesor inmediato
  - Por ejemplo, con **n = 4**, para los pares **1**, **3**, **4**, **5**, **8**, **10**, **12** y **14** definimos que para la clave **8** el sucesor inmediato es **8**, pero para **13** es **14** y para **15** es **1**

DHT circular



- Cada par sólo conoce los pares adyacentes
- Se conforma una red superimpuesta (overlay)

DHT circular

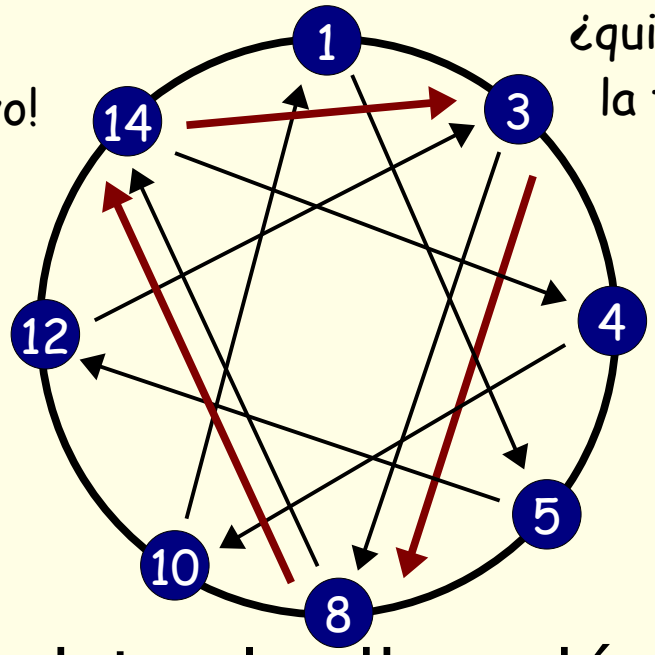


¿quién se encarga de la tupla cuya clave es "1110"?

esta consulta insumió 6 mensajes + 1 respuesta

- La resolución de una consulta insume **O(n)** mensajes para una red de **n** pares

DHT circular con atajos

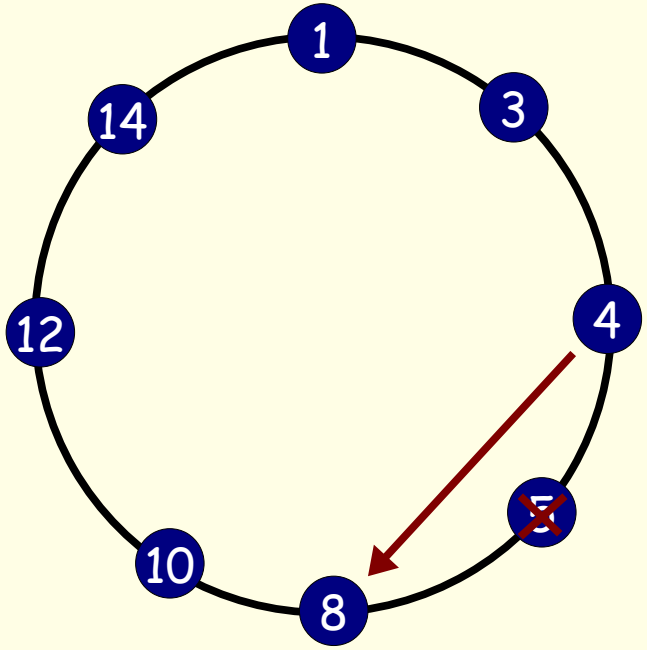


- Cada nodo registra la dirección de los nodos antecesor, sucesor y atajos
  - Con **`log n`** atajos, se reduce a **`O(log n)`** mensajes.

Actualización del DHT

- Dada la naturaleza dinámica del conjunto de pares, también se debe contemplar la incorporación y la eliminación de pares:
  - Para esto, cada par registra no sólo uno sino dos antecesores y dos sucesores
  - Periódicamente contacta al antecesor y al sucesor inmediato para ver si siguen en línea
  - En caso de no contestar, actualiza los sucesores y antecesores de manera apropiada
  - ¿Cómo se resolverá la incorporación de nuevos pares?

Actualización del DHT



- El nodo **4** detecta que **5** no contesta, hace **8** su sucesor inmediato y al sucesor de **8** su segundo sucesor. El nodo **8** también hace lo propio.

¿Preguntas?