

GRABACIÓN DE CLASE

Algoritmo de Ruteo Pastry

PASTRY Routing Table

NodeID: 10233102

Leaf Set

10233033	10233021	10233120	10331222
10233001	10233000	10233320	10233232

Routing Table

0	1	2	3
- 0 - 2212102	1	- 2 - 2301203	- 3 - 1203203
1 - 0 - ~	1 - 1 - 301233	1 - 2 - 230203	1 - 3 - 021022
10 - 0 - 32203	10 - 1 - 32102	10 - 2 - ~	10 - 3 - 23302
102 - 0 - 0230	102 - 1 - 1302	102 - 2 - 2302	102 - 3 - ~
1023 - 0 - 322	1023 - 1 - 000	1023 - 2 - 121	1023 - 3 - ~
10233 - 0 - 01	10233 - 1 - ~	10233 - 2 - 32	
0		102331 - 2 - 0	
-		2	

Neighborhood Set

13021022	10200230	11301233	31301233
02212102	22301203	31203203	33213321

Routing Algorithm

R_i the entry in routing table R at column i , $0 \leq i \leq 2b$, and row l , $0 \leq l < 128 / b$

L_i the i -th closest nodeID in the Leaf Set L

D_i the value of the i 's digit in the key D

$shl(A, B)$ the length of the prefix shared among A and B in digits

```

If (  $L_{\lfloor |L|/2 \rfloor} \leq D \leq L_{\lfloor |L|/2 \rfloor + 1}$  )
  // D is within range of our Leaf Set
  Forward to  $L_i$  such that  $|D - L_i|$  is minimal
Else {
  // Use the routing table
   $l = shl(D, A)$ ;
  if (  $R_l^{D_l} \neq \text{null}$  )
  {
    Forward to  $R_l^{D_l}$ 
  }
  Else {
    // rare cases
    forward to  $T \in L \cup R \cup M$  such that  $shl(T, D) \geq l$ ,  $|T - D| < |A - D|$ 
  }
}

```

Handwritten notes:

- Cant. de Digits compartidos con Destino
- 102, 2, 1, 3, 3
- cuantos digitos comparte
- [3, 2]
- Siguele now al que le pasa el mst.
- nosos con "buena" conexión vel. más rápida.

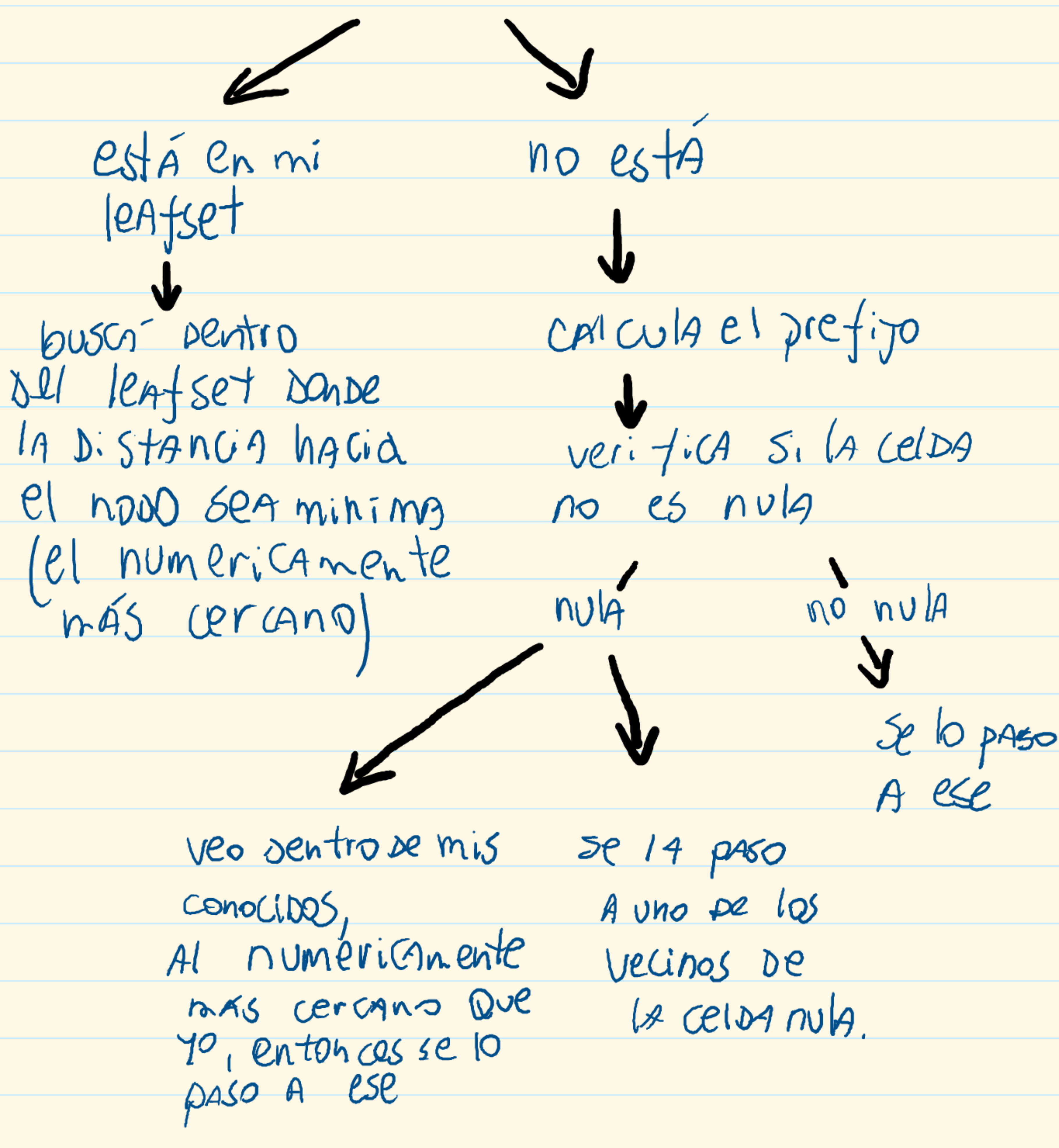
primero verifica cuantos digitos tiene en común con el destino final. (en base a su hash)

Nº de Saltos

$$O(\log(N))$$

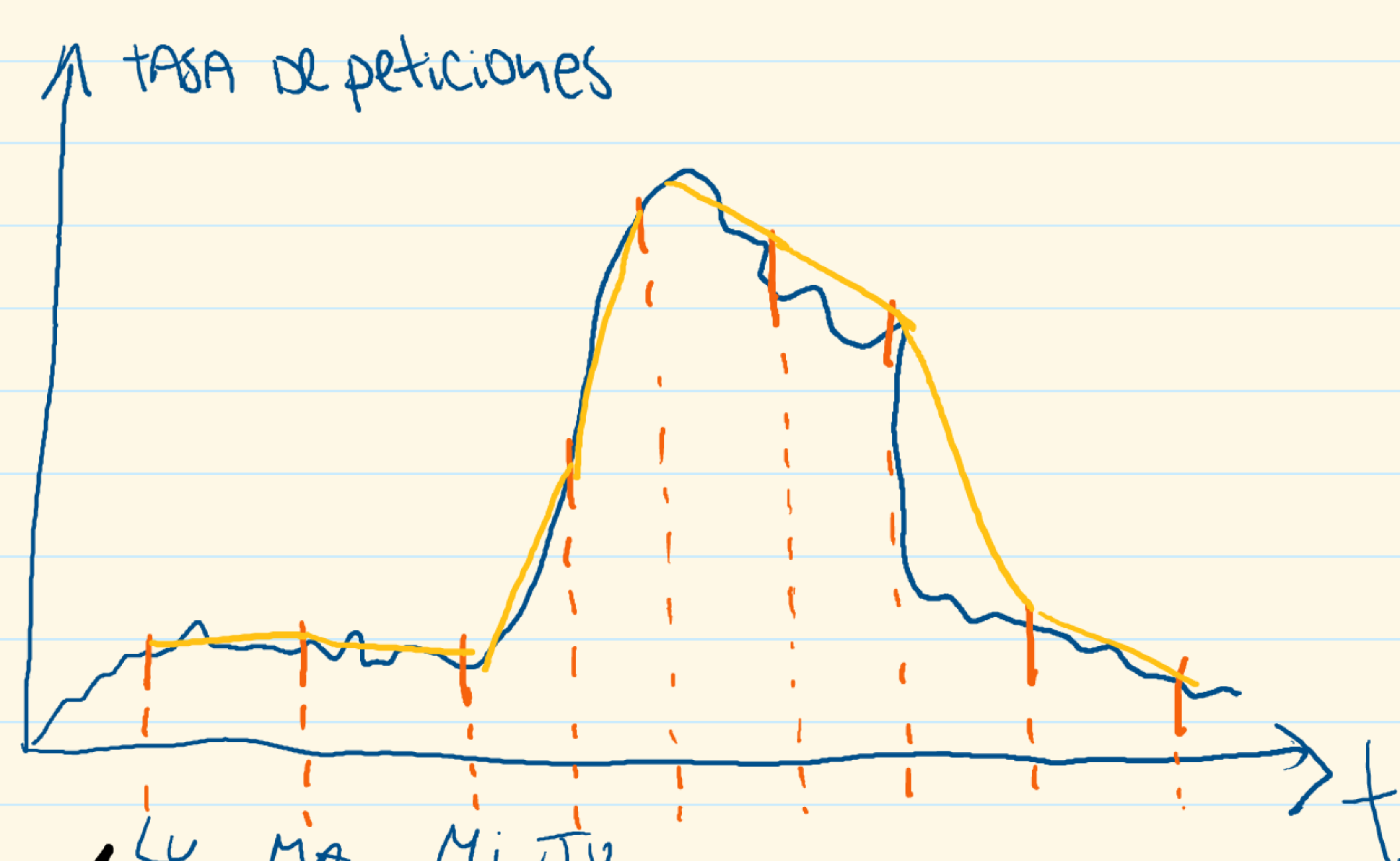
↑
cant. total de nodos

Ruteo



Cuando existe mucho furor por "superman" cuando sale, el balanceo de carga natural ya no me funciona

✗ Generar + copias (Aumentando el leafset)
eso trae consigo + mst: de tipo KEEP_ALIVE costo +



✓ Una solución sería verificar cada Δt la pendiente de la tasa de peticiones

¿pero como repartimos las copias?

el hash del hash del hash...

$$h_1 = \text{Sha}(\text{"superman"})$$

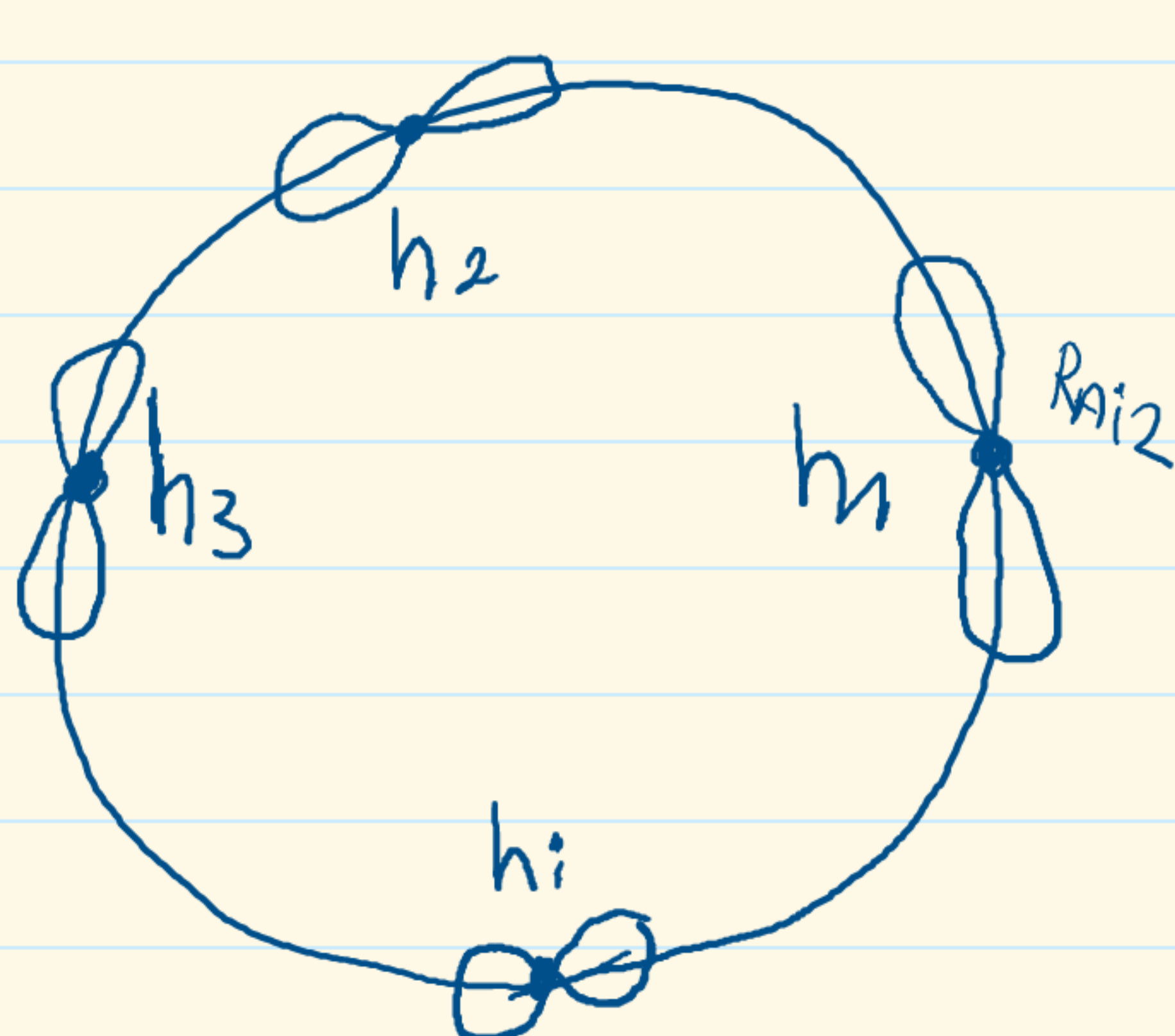
$$h_2 = \text{Sha}(h_1)$$

$$h_3 = \text{Sha}(h_2)$$

⋮

$$h_i = \text{Sha}(h_{i-1}) = \text{Sha}^i(\text{"superman"})$$

las replicamos en otros leafset, así no tenemos costo de mantención del nullo.



n leafset.
para replicar
copias de superman.

para que exista un balanceo extra de carga necesitamos hacer un Random entre 1 y K donde K será el número de leafset que tiene a "superman"

CADA nodo debe tener guardado el valor K de copias de "superman"

$$L = \text{Sha}(\text{"superman"} + \text{"niveles"})$$

para guardar estos datos, es como agregarle un salt. (como en las contraseñas)