

**Lezione n.6**  
**Peer-to-Peer Systems**  
**and Applications**  
**Capitolo 8**

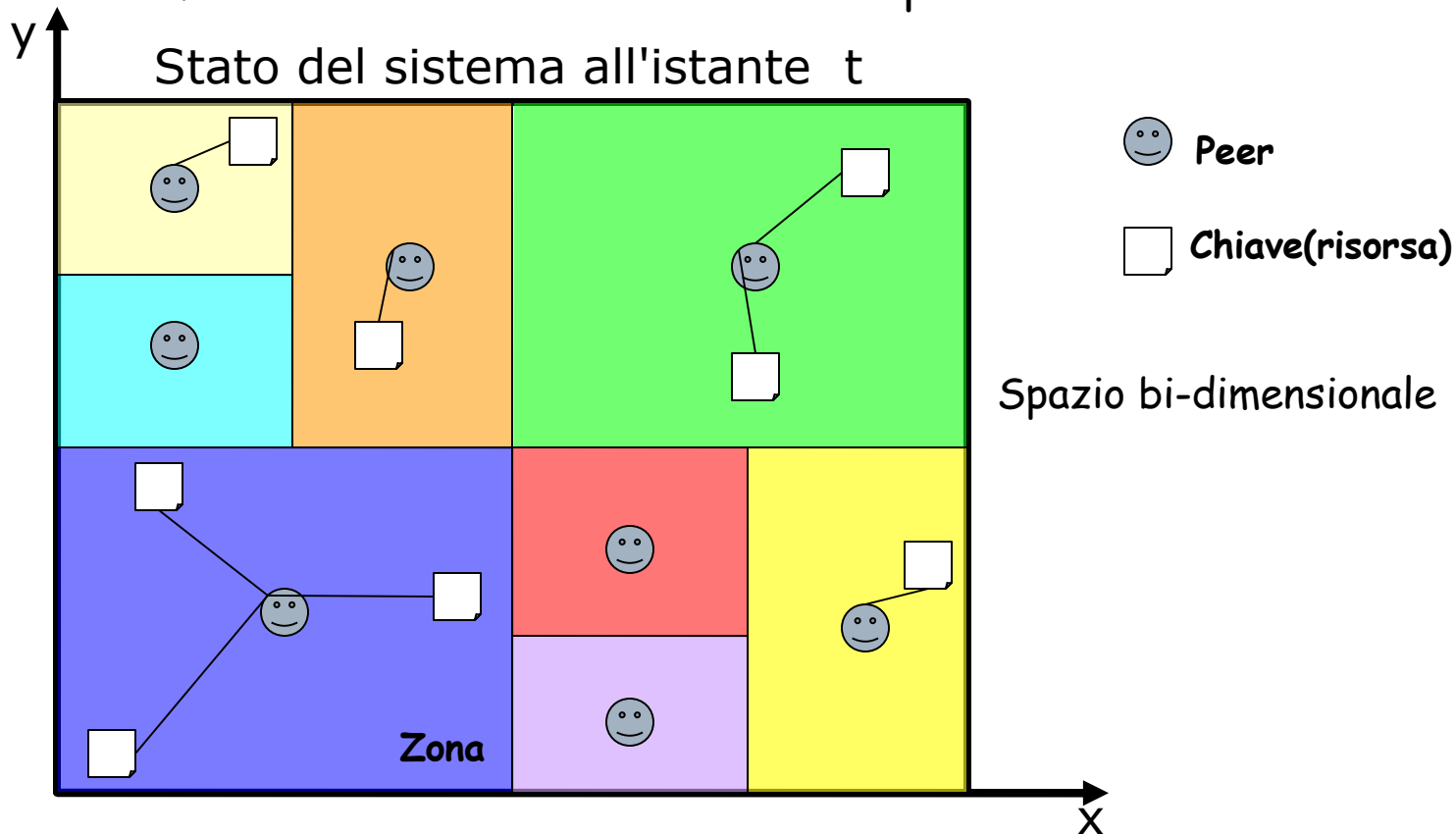
**CAN: Content Addressable Network**  
**Laura Ricci**

# CONTENT ADDRESSABLE NETWORK

- Proposto da ricercatori dell'Università della California, Berkeley e dell'AT&T
- Materiale: *A Scalable Content-Addressable Network*, Sylvia Ratnasamy, P.Francis, M.Handley, R. Karp, S. Shenker " proceedings Applications, Technologies, Architectures, and Protocols for Computer Communication
- Proposto nel 2001
- Non disponibile per il download

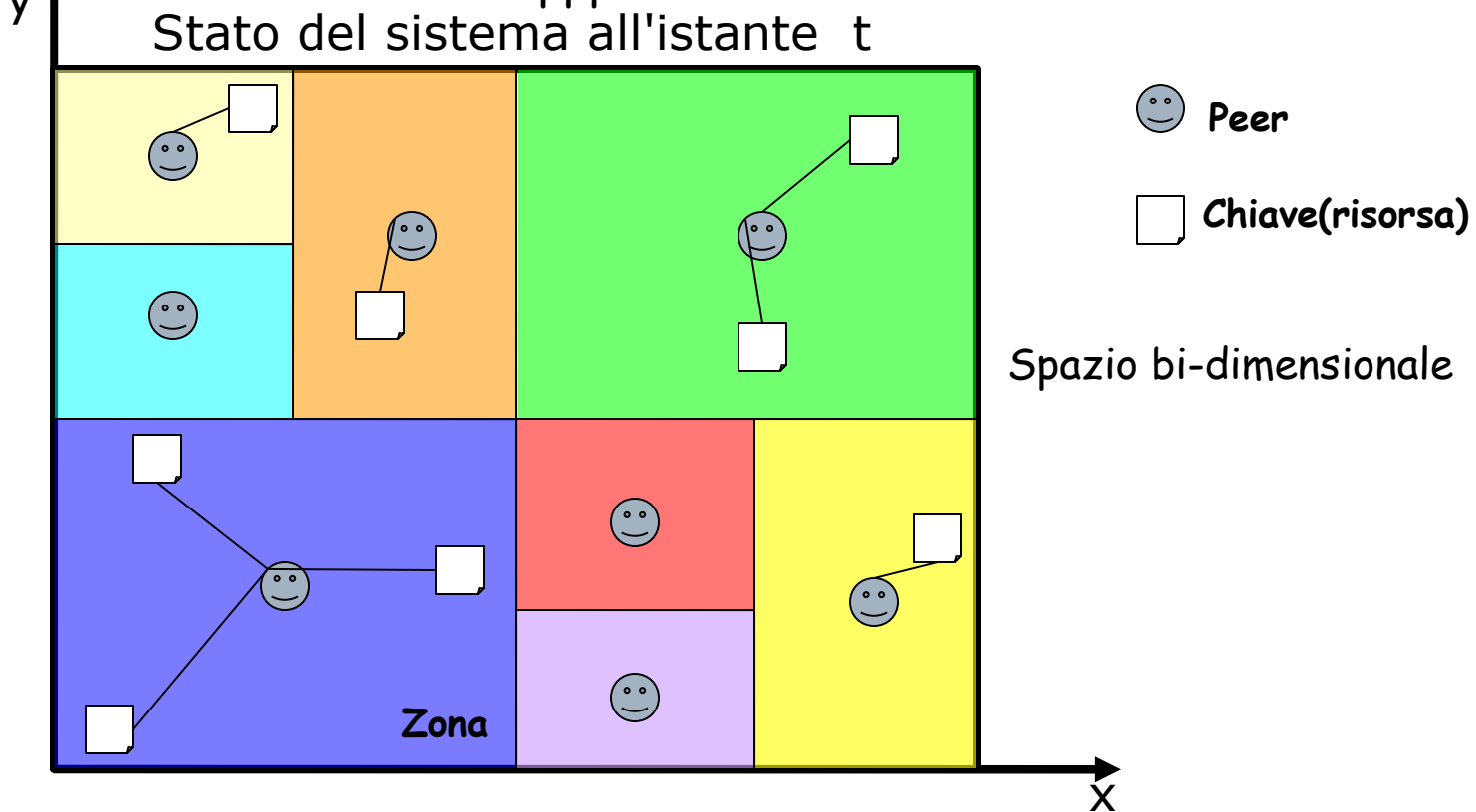
# CAN: IDEE DI BASE

- associa ad ogni nodo e ad ogni informazione un punto in uno spazio cartesiano **d-dimensionale, con richiuse toroidali**
- ogni peer gestisce le chiavi associate ad una zona dello spazio



# CAN: IDEE DI BASE

- Lo spazio è diviso tra i nodi e viene interamente coperto dai peer
- Ogni nodo copre un'area quadrata oppure un'area rettangolare in cui il rapporto tra base ed altezza è 1:2 oppure 2:1



# CONTENT ADDRESSABLE NETWORK

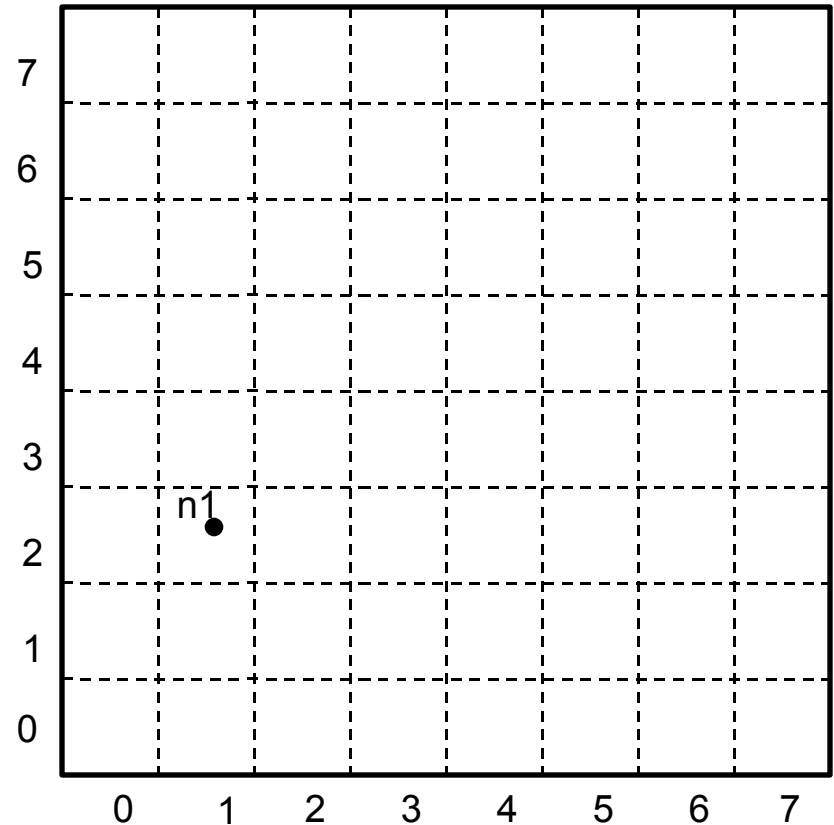
- Obiettivi
  - Scalabilità a centinaia di migliaia di nodi
  - Gestione efficiente di inserimento e caduta di nodi
- Caratteristiche per una rete a  $d$  dimensioni
  - dimensione della Tabella di Routing  $O(d)$
  - numero medio di passi necessari per individuare un'informazione (routing hops)  $d * n^{1/d}$ , dove  $n$  è il numero totale di nodi

# CAN: CONTENT ADDRESSABLE NETWORK

- Per ricavare le coordinate di un nodo o di un dato nello spazio  $d$ -dimensionale
  - Applicazione di una funzione hash all'identificatore del nodo o al dato
  - La funzione hash mappa il nodo/dato in una sequenza di bits
  - La sequenza di bits ottenuta è partizionata in  $d$  sottosequenze. Ogni sottosequenza rappresenta una coordinata nello spazio a  $d$  dimensioni
- Le coordinate in ogni dimensione variano in un intervallo di valori
- Tutte le operazioni sono calcolate in modulo rispetto alla valore più grande della coordinata in una dimensione
- Lo spazio degli indirizzi CAN può essere rappresentato come un toro a  $d$  dimensioni

# CAN: PARTIZIONAMENTO DELLO SPAZIO

- Il primo nodo gestisce l'intero spazio delle chiavi
- **Esempio:** Il nodo n1 mappato nel punto (1, 2) è il primo nodo che entra nel sistema e ad esso viene associato l'intero spazio bi-dimensionale

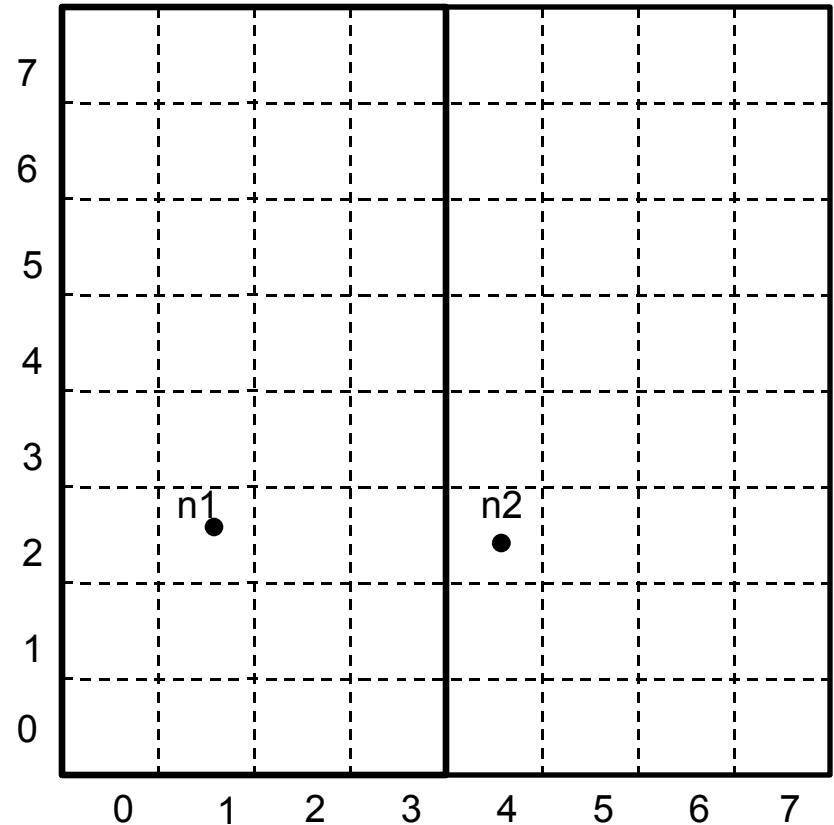


# CAN: PARTIZIONAMENTO DELLO SPAZIO

Il nodo  $n2:(4, 2)$  si unisce alla rete CAN

$\Rightarrow$

lo spazio viene partizionato tra  $n1$  ed  $n2$



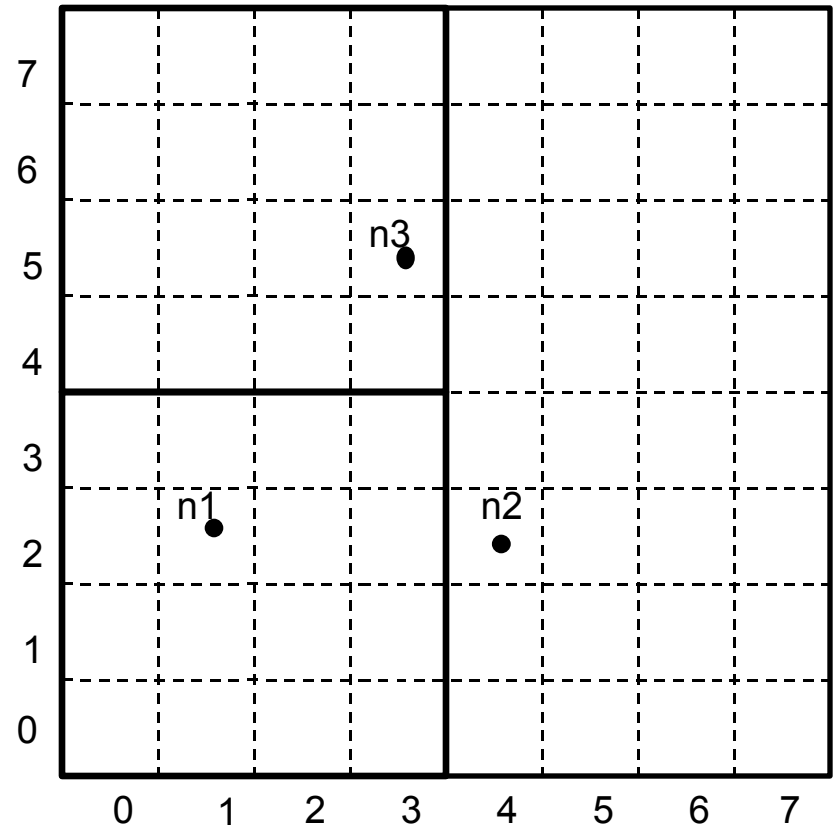


# CAN: PARTIZIONAMENTO DELLO SPAZIO

il nodo  $n3:(3, 5)$  si unisce alla rete

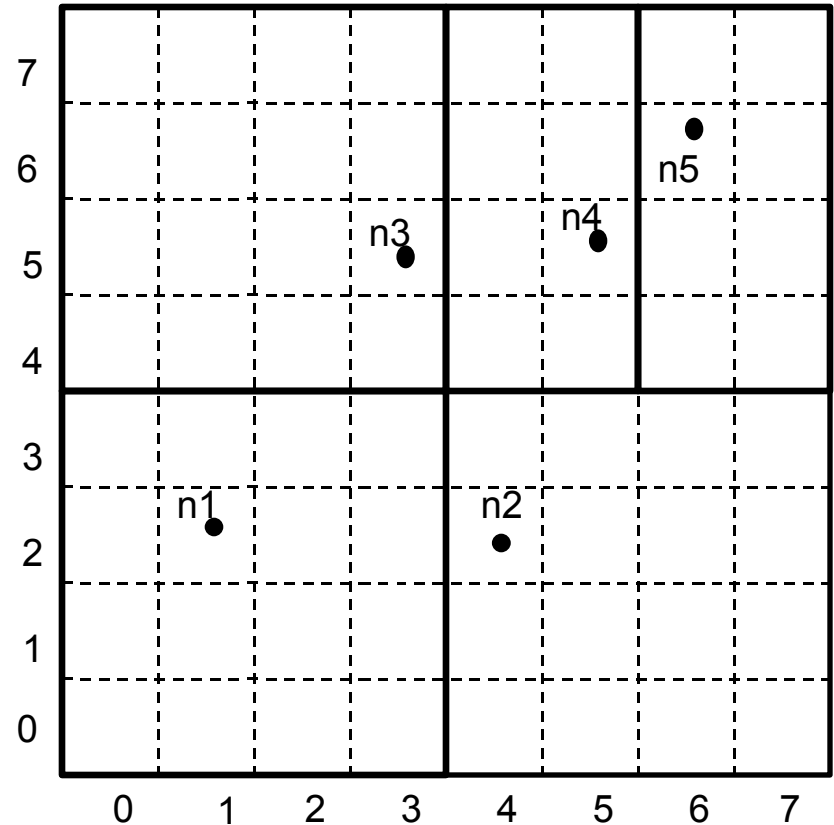
⇒

lo spazio viene diviso tra  $n1$  ed  $n3$



# CAN: PARTIZIONAMENTO DELLO SPAZIO

$n4:(5, 5)$  and  $n5:(6,6)$  si uniscono alla rete  
CAN



# CAN: ASSOCIAZIONE DELLE CHIAVI AI NODI

Ogni chiave viene memorizzato dal nodo che gestisce la zona in cui è stato mappata la chiave stessa

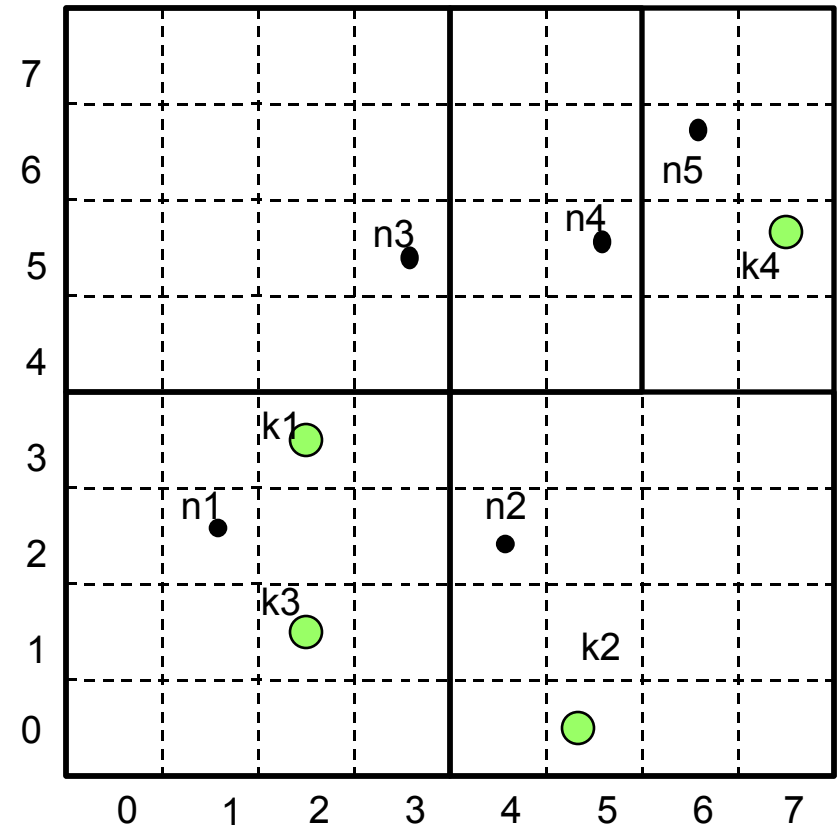
## Nodi:

$n1:(1, 2)$ ;  $n2:(4, 2)$ ;  $n3:(3, 5)$ ;

$n4:(5, 5)$ ;  $n5:(6, 6)$

## Chiavi

$k1:(2, 3)$ ;  $k2:(5, 1)$ ;  $k3:(2, 1)$ ;  $k4:(7, 5)$ ;



# CAN: ASSOCIAZIONE DELLE CHIAVI AI NODI

Ogni chiave viene memorizzato dal nodo che gestisce la zona in cui è stato mappata la chiave stessa

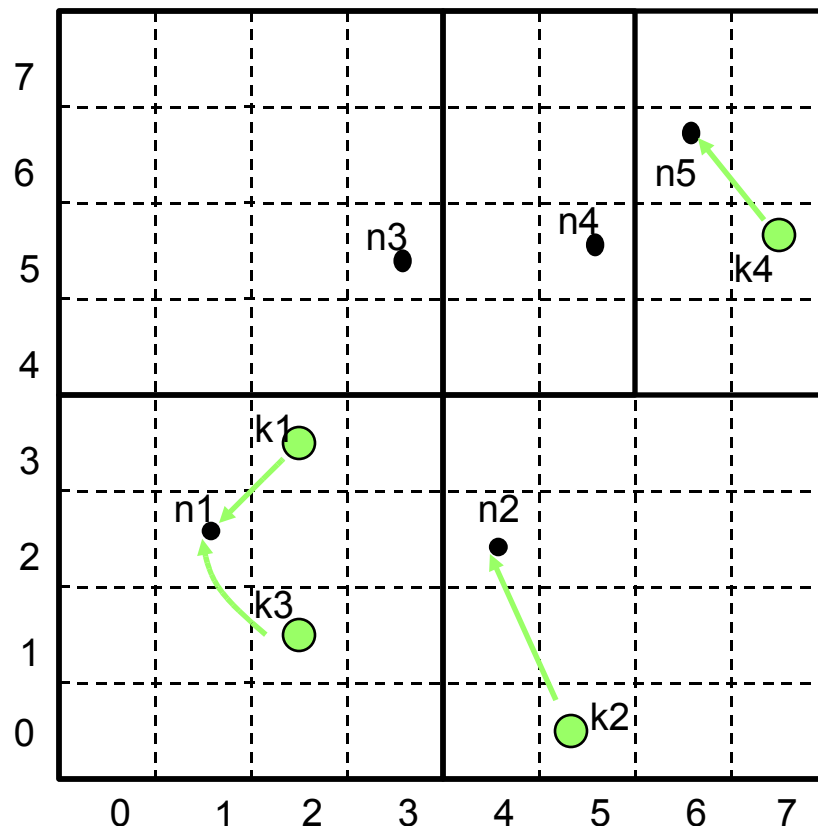
## Nodi:

$n1:(1, 2); n2:(4,2); n3:(3, 5);$

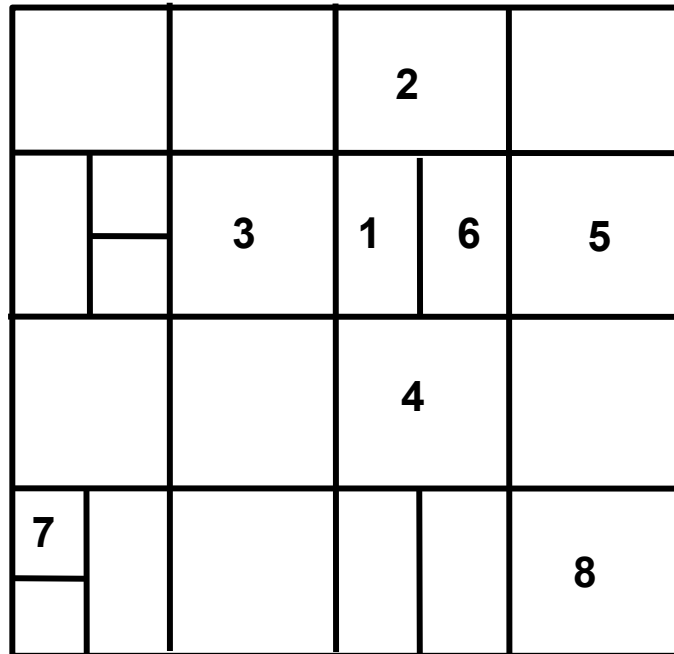
$n4:(5,5); n5:(6,6)$

## Chiavi

$k1:(2,3); k2:(5,1); k3:(2,1); k4:(7,5);$



# CAN: DEFINIZIONE DEI VICINI



- Due nodi sono considerati vicini se e solo se le rispettive zone "sono in contatto" in più di un punto
  - due zone in uno spazio a  $D$  dimensioni sono in contatto se i loro lati sono adiacenti rispetto ad una dimensione e si sovrappongono rispetto alle rimanenti  $d-1$  dimensioni

- Ad esempio il nodo 5 è adiacente al nodo 6: i lati delle rispettive zone sono adiacenti rispetto all'asse  $x$  e si sovrappongono lungo l'asse  $y$
- i nodi 4 e 5 non sono vicini: i loro lati sono adiacenti in entrambe le dimensioni, ma non si sovrappongono

# CAN: TABELLE DI ROUTING

		2		
	3	1	6	5
		4		
7				8

- $D=2$
- vicini di 1: 2,3,4,6
- vicini di 6: 1,2,4,5
- considerare **le richiature**:  
7 ed 8 sono considerati vicini

**Tabelle di Routing** contiene un elemento per ogni vicino P e memorizza

- l'indirizzo IP e la porta di P
- le coordinate della zona che P gestisce

# CAN: TABELLA DI ROUTING

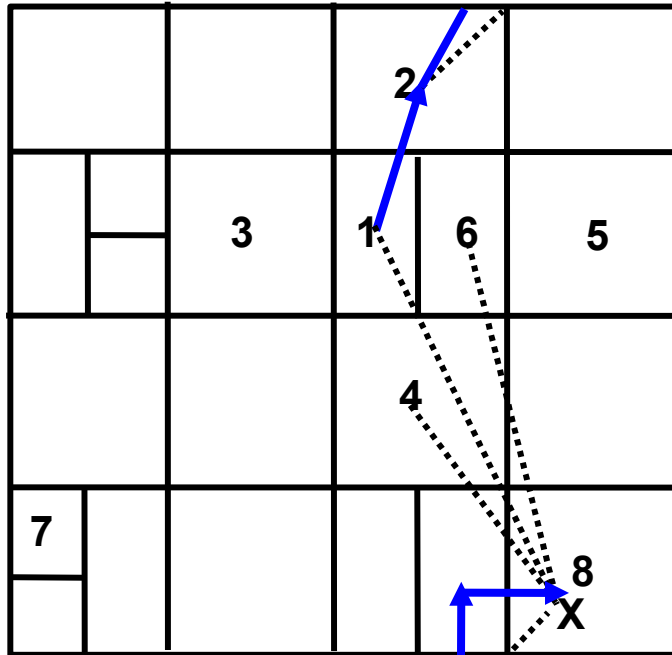
- Il valore restituito dalla funzione hash è considerato un punto in uno spazio cartesiano a  $D$  dimensioni
- Ogni nodo è responsabile di una zona "cubo a  $D$  dimensioni"
- Lo spazio a  $D$  dimensioni è uno spazio logico e non presenta alcuna relazione con lo struttura della rete fisica sottostante
- Dimensione della tabella di routing: se si considera uno spazio a  $D$  dimensioni diviso in  $n$  zone della stessa ampiezza, ogni nodo deve mantenere  $2D$  vicini
- Approssimazione: distribuzione uniforme dei punti nello spazio. In questo caso, al crescere del numero di nodi del sistema, l'informazione richiesta per il routing è  $O(D)$ .

# CAN: IL ROUTING

- Ogni messaggio CAN contiene le coordinate di un punto  $P$  destinazione ad esempio le coordinate associate dalla funzione hash ad una chiave.
- Ogni nodo instrada il messaggio verso il nodo che gestisce la zona che contiene  $P$ .
- Routing: Approssimazione di "seguire il segmento che collega sorgente e destinazione"
- Algoritmo di routing (*strategia greedy*):
  - Il nodo  $n$  inoltra un messaggio diretto verso il punto  $P$  di coordinate  $(c_1, c_2, \dots, c_D)$  verso il vicino che gestisce la zona più vicina a  $P$ .
  - La distanza tra una zona ed il punto  $P$  viene misurata rispetto al punto centrale della zona.



# CAN: IL ROUTING



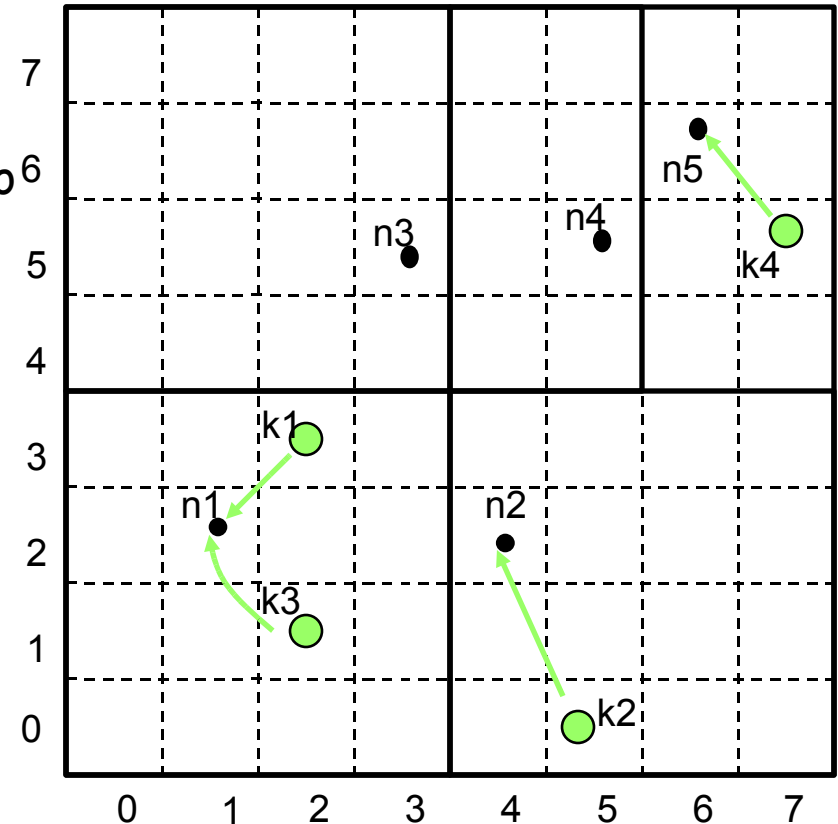
• **Esempio:** il nodo che gestisce la regione 1 deve ricercare la chiave X

- controlla le distanze da X di tutte le zone associate ai vicini
- inoltra il messaggio al nodo 2 (considerare le richiuse toroidali)

• la distanza cartesiana dal punto destinazione decresce ad ogni hop

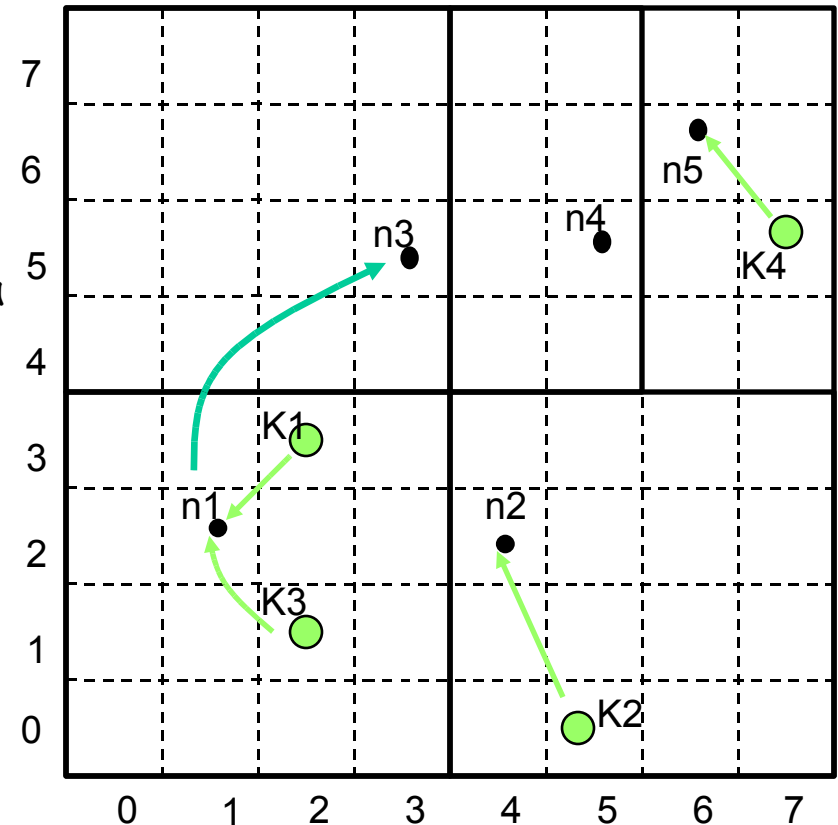
# CAN: IL ROUTING

- Ogni nodo conosce i suoi vicini nello spazio a d dimensioni
- La query viene inoltrata al vicino che possiede la zona le cui coordinate risultano più vicine a quelle del dato ricercato
- Esempio:  
n1 ricerca k4



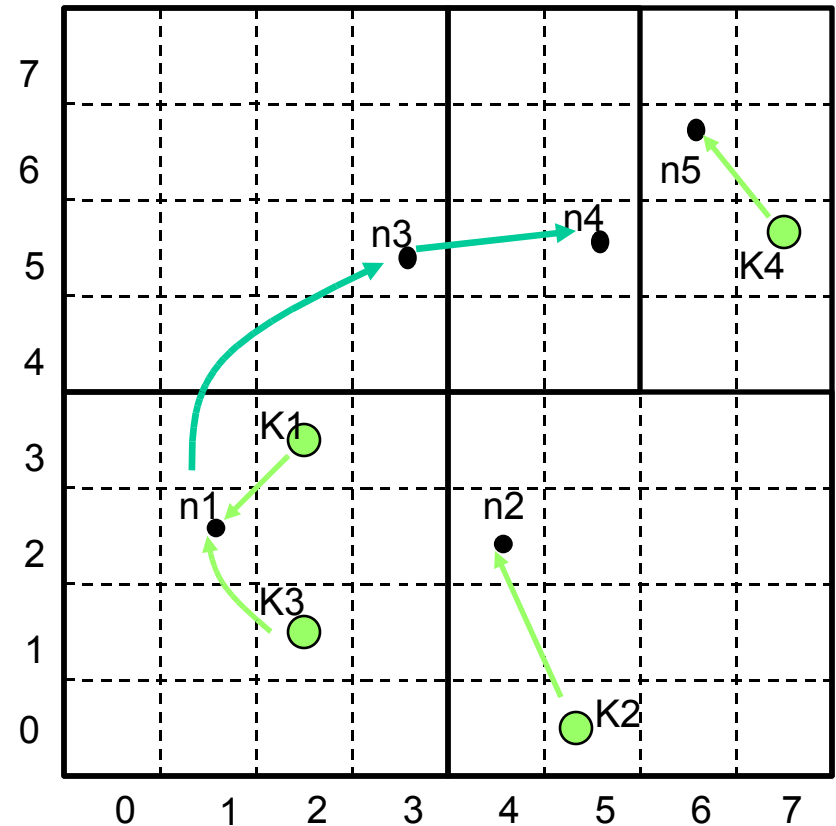
# CAN: IL ROUTING

- Ogni nodo è a conoscenza dei suoi vicini nello spazio a  $d$  dimensioni
- La query viene inoltrata al nodo che risulta più vicino all'id della query
- Esempio:
  - $n1$  ricerca  $k4$



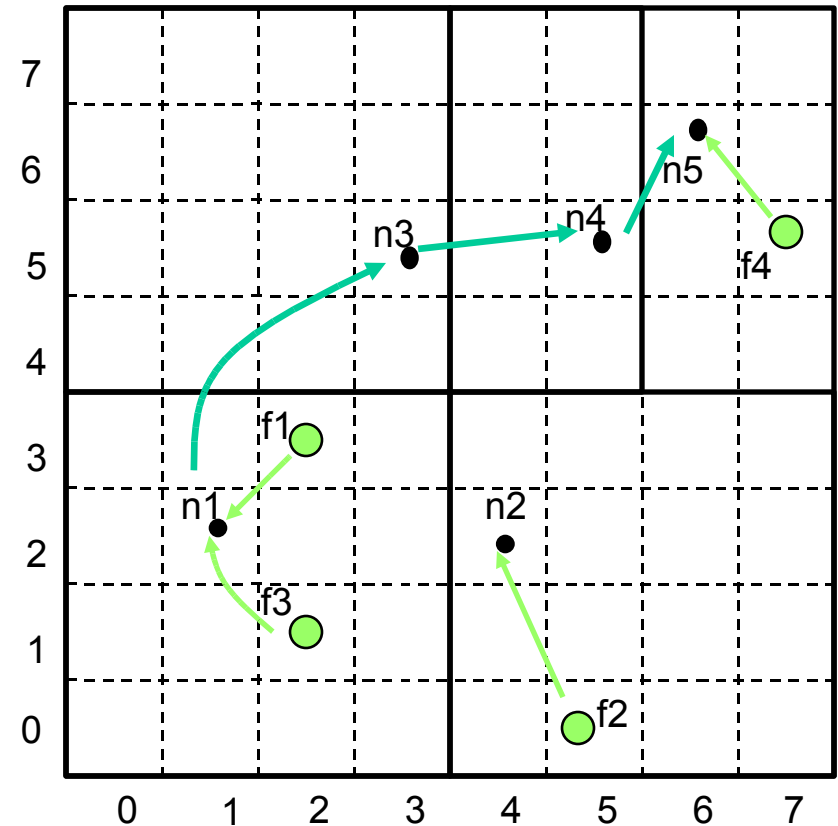
# CAN: IL ROTUING

- Ogni nodo è a conoscenza dei suoi vicini nello spazio a d dimensioni
- La query viene inoltrata al nodo che risulta più vicino all'id della query
- Esempio:  
n1 ricerca K4



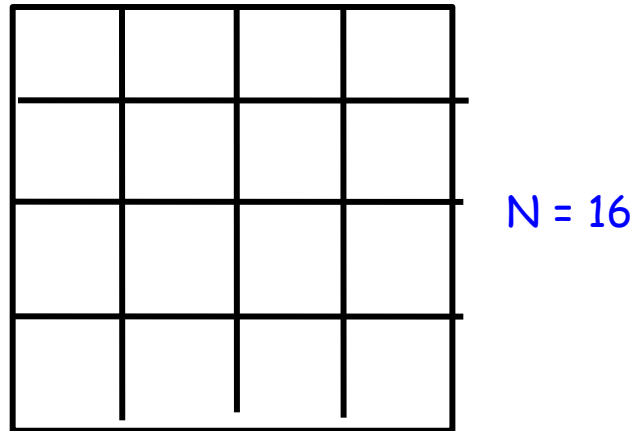
# CAN: IL ROUTING

- Ogni nodo è a conoscenza dei suoi vicini nello spazio a d dimensioni
- La query viene inoltrata al nodo che risulta più vicino all'id della query
- Esempio:  
n1 ricerca K4



# CAN: COMPLESSITA' DEL IL ROUTING

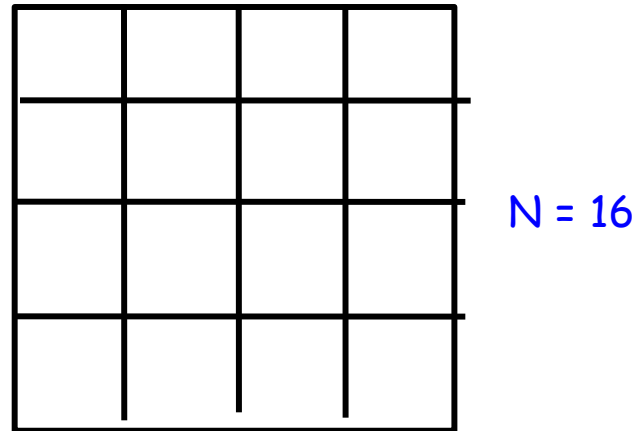
- Complessità del routing  $O(D \cdot N^{1/D})/4$ . se lo spazio è equamente diviso in n zone
- Consideriamo uno spazio bi-dimensionale decomposto in quadrati di uguale dimensione



- Consideriamo uno spazio bi-dimensionale decomposto in quadrati di uguale dimensione
- Ogni dimensione ha  $N^{1/D}$  nodi ( $16^{1/2} = 4$ )
- Nel **caso pessimo** percorro  $\frac{1}{2} * N^{1/D}$  passi in ognuna delle dimensioni ( $\frac{1}{2}$  a causa delle richiuse), quindi  $D * \frac{1}{2} * N^{1/D}$

# CAN: COMPLESSITA' DEL IL ROUTING

- Complessità del routing  $O(D \cdot N^{1/D})/4$ . se lo spazio è equamente diviso in n zone:



- Per il caso medio devo considerare 0 passi nel caso ottimo, oppure 1 passo,.....  
oppure  $D \cdot \frac{1}{2} \cdot N^{1/D}$  nel caso pessimo
- In media  $(\sum i, i=0..k) / k+1$ , dove  $k = D \cdot \frac{1}{2} \cdot N^{1/D}$
- $(\sum i, i=0..k) / k+1 = k(k+1)/2 \cdot k+1 = k/2$
- Poichè  $k = D \cdot \frac{1}{2} \cdot N^{1/D}$ , si ottiene il risultato  $O(D \cdot N^{1/D})/4$ .

# CAN: INSERZIONE DI NUOVI NODI

Quando un nuovo nodo  $n$  si unisce ad una rete  $R$  di nodi CAN:

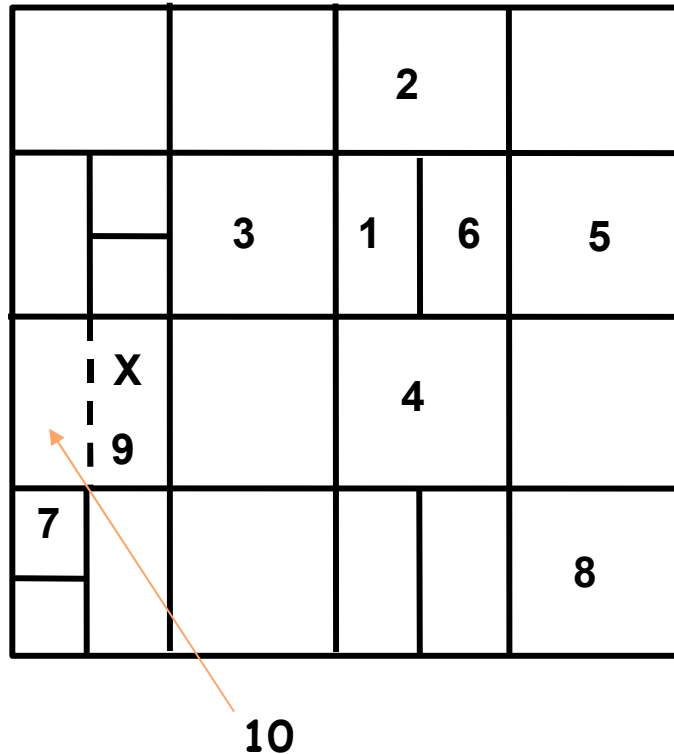
- ricerca un **nodo di bootstrap**  $B$  già esistente all'interno di  $R$  (esempio: si associa un **nome simbolico ad una rete CAN**, il DNS restituisce l'IP di uno o più nodi CAN)
- **sceglie, in modo casuale, un punto  $P$**  nello spazio CAN o applica una funzione hash al proprio indirizzo IP + porta
- invia un messaggio di **JOIN**, utilizzando  $P$  come chiave. Il messaggio viene recapitato al **nodo  $C$**  che gestisce la zona che contiene  $P$
- il messaggio di **JOIN** viene spedito **mediante il nodo  $B$**
- il messaggio raggiunge  $C$  mediante il meccanismo di routing di CAN



# CAN: INSERZIONE DI NUOVI NODI

- Il nodo  $C$  che riceve un messaggio di JOIN dal nodo  $n$  divide la propria zona in due parti
  - il primo taglio viene effettuato lungo la prima dimensione
  - se l'ultimo taglio è effettuato lungo la direzione  $i < D$ , il successivo taglio avviene lungo la dimensione  $i+1$ -esima (modulo  $D$ )
  - Se  $D=2$ , si divide prima lungo l'asse delle  $x$ , poi lungo quello delle  $y$
- $C$  tiene metà dello spazio per se e cede la restante parte a  $n$
- $C$  cede le chiavi eventualmente mappate sullo spazio ceduto a  $n$
- Osservazione: la probabilità che una zona venga scelta dipende dalla sua dimensione

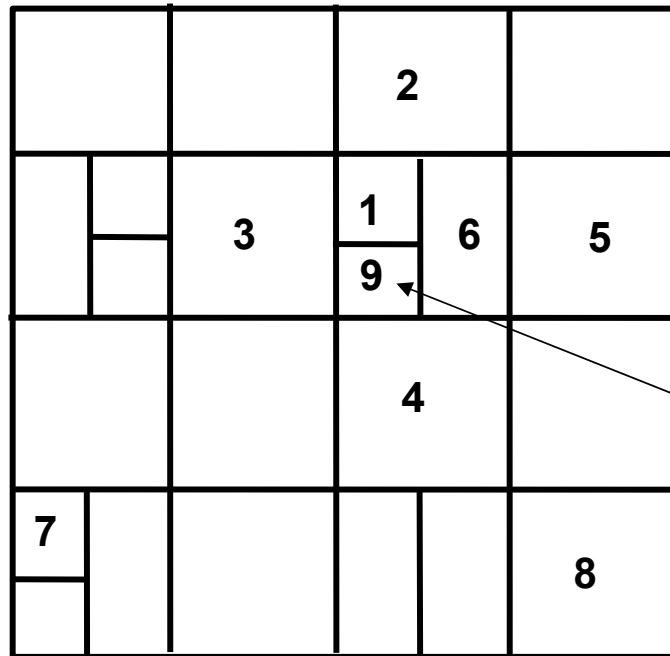
# CAN: INSERZIONE DI NUOVI NODI



- Identifichiamo i nodi con i numeri 1,...,10, solo per identificarli
- Tenere presente che CAN non assegna un identificatore al nodo, ma **una zona**
- il nuovo nodo 10 sceglie, in modo casuale, **un punto, ad esempio X**
- X è contenuto nella zona gestita dal nodo 9
- 9 cede metà della propria zona a 10
- Se ci sono chiavi mappate su quella parte della zona assegnata al 9, queste vengono trasferite al 10

# CAN: AGGIORNAMENTO VICINI

- Il nuovo nodo  $n$  riceve le informazioni relative ai suoi vicini dal nodo  $C$  che ha ceduto parte della propria zona
- i vicini di  $n$  sono un sottoinsieme dei vicini di  $C$
- $Z$  aggiorna i propri vicini in base alle coordinate della zona ceduta



Dopo l'inserimento del nuovo nodo

• vicini di 9 = {1,3,6,4}

• vicini di 1 = {2,3,4,9}

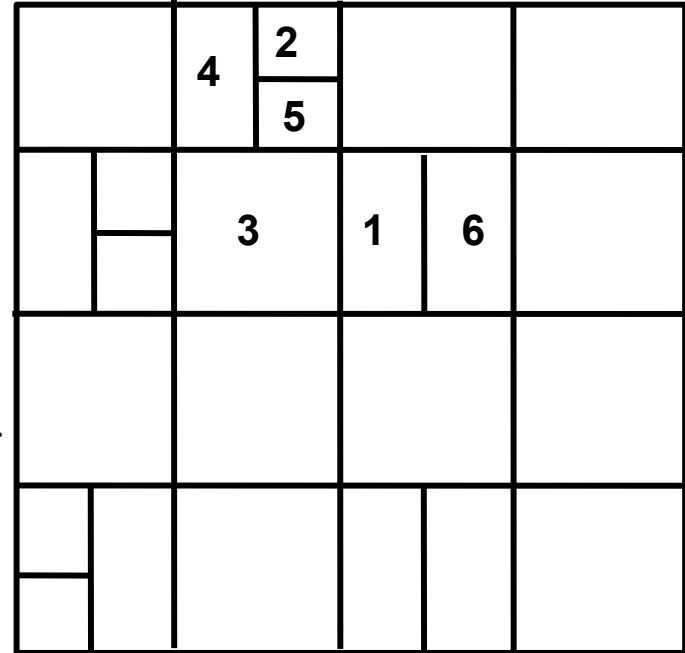
**Nuovo nodo 9**

# CAN: USCITA DI NODI DALLA RETE

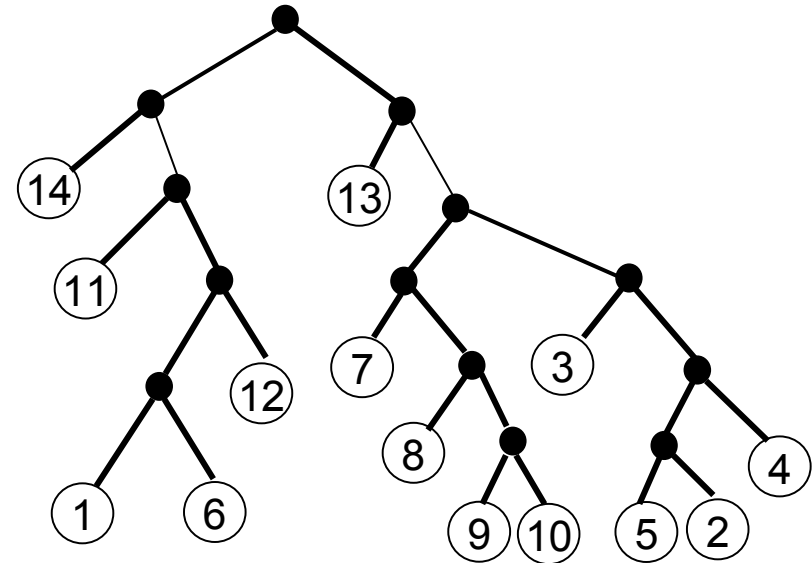
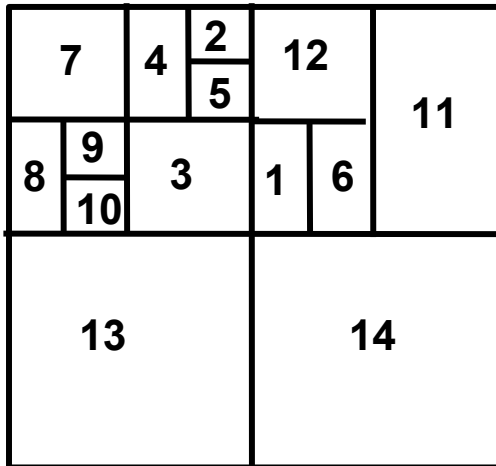
- Quando un nodo  $n$  decide volontariamente di uscire dalla rete, la gestione della zona  $Z$  controllata da  $n$  deve essere passata ad un nodo vicino (takeover node)
- $n$  consegna autonomamente le informazioni associate a  $Z$  ad un nodo  $n'$  che controlla una zona vicina  $Z'$ . Le informazioni trasmesse ad  $n'$  sono:
  - (key,value) mappate su  $Z$
  - identificazione dei vicini di  $n$  (zone + indirizzi IP)
- Se possibile, le zone di  $n$  ed  $n'$  vengono fuse in modo da produrre una zona più ampia che mantenga la struttura CAN
- Altrimenti,  $n'$  prende in carico sia  $Z$  che  $Z'$ , momentaneamente senza effettuare alcuna fusione.

# CAN: USCITA DI NODI DALLA RETE

- La fusione delle zone deve mantenere la struttura topologica di CAN
- **Esempio:** 6 lascia la rete. La sua zona può essere fusa con quella controllata da 1
- **Esempio:** 3 lascia la rete. La sua zona non può essere fusa con le zone adiacenti (4,5). 5 prende in consegna la zona 3.



# CAN: PARTITION TREE

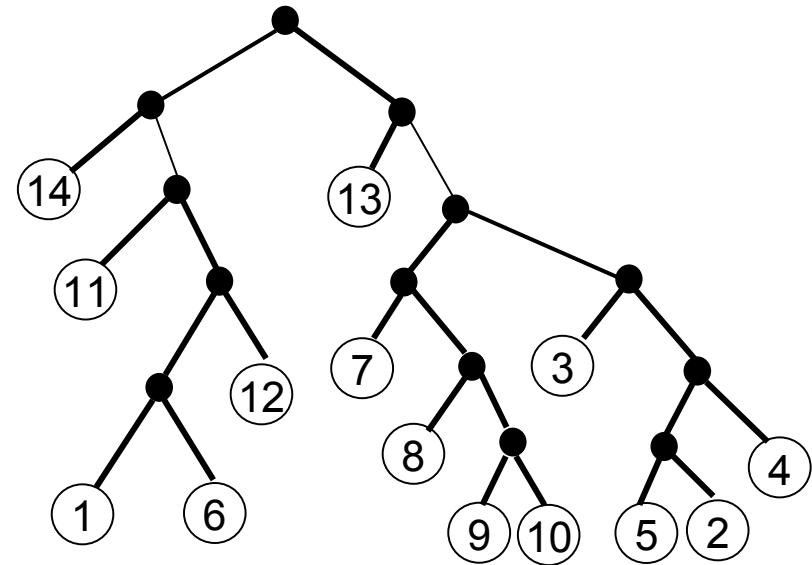


Il partizionamento dello spazio CAN puo' essere rappresentato mediante un **albero binario** in cui

- le **foglie** rappresentano le zone assegnate ai nodi
- i **nodi interni** rappresentano zone che non esistono più, in quanto decomposte
- i **figli di un nodo** rappresentano le due zone in cui è stata decomposta una zona

# CAN: INDIVIDUAZIONE DEL TAKE OVER NODE

7	4	2	12		11
		5			
8	9	3		1	6
	10				
13			14		

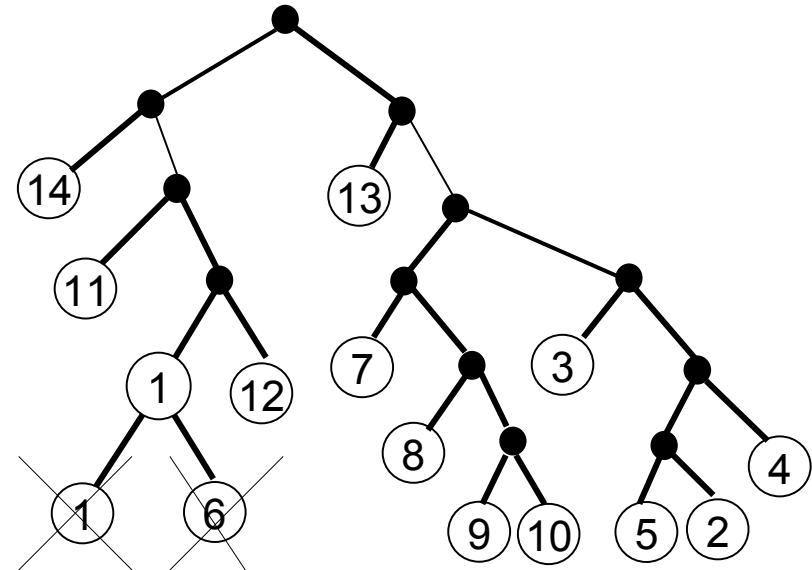


Caso 1: la foglia  $s$  viene rimossa, il fratello di  $s$  è una foglia  $t$

- $s$  e  $t$  vengono fuse
- il nodo padre  $p$  diventa un nodo foglia
- la zona corrispondente a  $p$  viene presa in carico dal nodo CAN che in precedenza gestiva  $t$

# CAN: INDIVIDUAZIONE DEL TAKE OVER NODE

7	4	2	12		11
		5			
8	9	3		1	6
	10				
13			14		



Caso 1: la foglia  $s$  viene rimossa, il fratello di  $s$  è una foglia  $t$

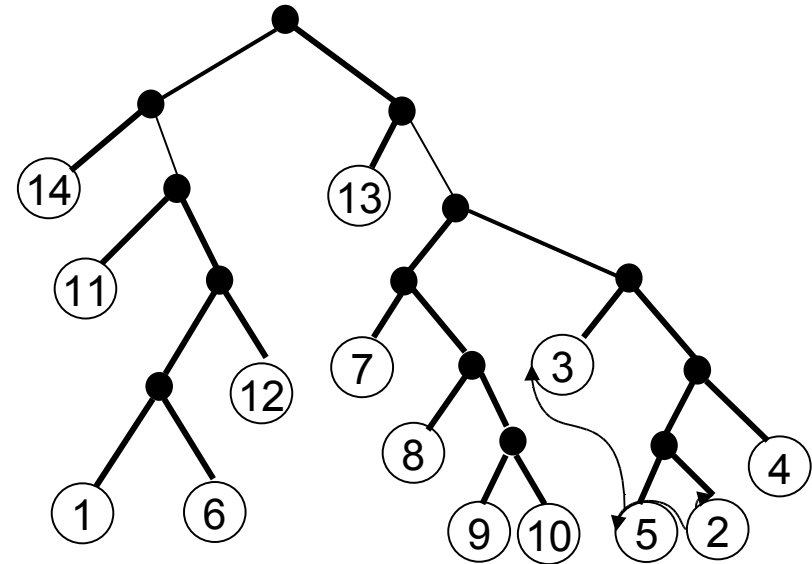
Esempio: 6 lascia la rete CAN

- le zone 1 e 6 vengono fuse
- 1 prende in consegna la zona risultante



# CAN: INDIVIDUAZIONE DEL TAKE OVER NODE

7	4	2	12	11
		5		
8	9	3	1	6
	10			
13			14	



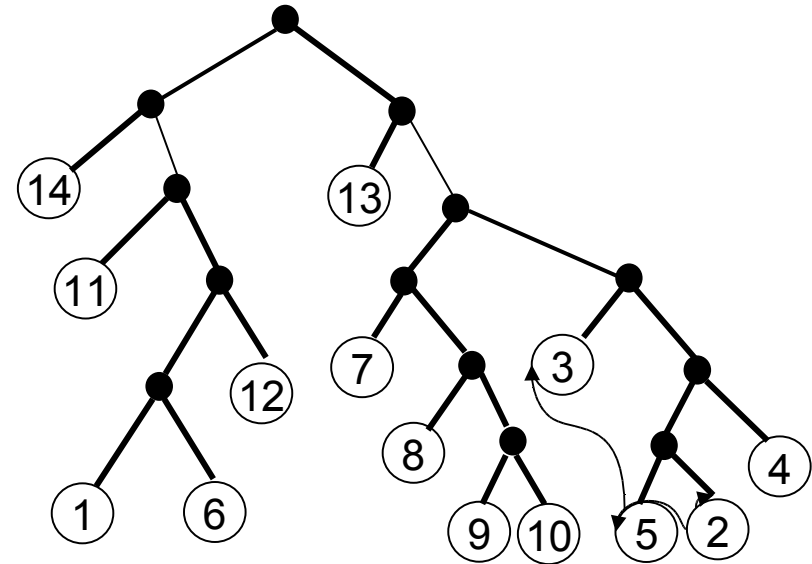
Caso 2: la foglia  $s$  viene rimossa, il fratello  $t$  di  $s$  non è una foglia

- si effettua una visita **depth first (DFS)** nel sottoalbero radicato in  $t$  fino ad individuare una foglia  $z$  che ha come fratello un nodo foglia
  - se  $s$  è il fratello sinistro di  $t$  la DFS visita per primo il sottoalbero sinistro e poi quello destro, viceversa se  $s$  è il fratello destro di  $t$ .
- Il nodo corrispondente a  $z$  prende in consegna la zona corrispondente ad  $s$
- Il nodo corrispondente al fratello di  $z$  prende in consegna la zona gestita da

Z

# CAN: INDIVIDUAZIONE DEL TAKE OVER NODE

7	4	2	12	11
		5		
8	9	3	1	6
	10			
13			14	



Caso 2: la foglia *s* viene rimossa, il fratello *t* di *s* non è una foglia

Esempio: 3 lascia la rete CAN

- Si effettua una visita DFS a partire dal fratello di 3 nell'albero, visitando per primo il sottoalbero sinistro, fino a trovare la foglia 5
- Il nodo corrispondente a 5 prende in consegna la zona 3
- La zona 5 viene presa in consegna dal nodo che gestisce la zona 2

# CAN: INDIVIDUAZIONE DEL TAKE-OVER NODE

- La procedura precedente utilizza una conoscenza globale dell'albero bianrio che rappresenta il partizionamento dello spazio
- E' necessario definire un algoritmo distribuito che utilizzi solamente l'informazione locale di un nodo
- Occorre sfruttare una struttura che indichi la adiacenza tra zone contigue
- **Esercizio: Progettare un algoritmo distribuito che implementi la soluzione illustrata nei lucidi precedenti, realizzando in modo distribuito la DFS illustrata nei lucidi precedenti**

# CAN: FALLIMENTO DI NODI

- CAN non prevede meccanismi di replicazione. Le informazioni memorizzate sul nodo fallito sono perse definitivamente
- Si suppone che l'applicazione che utilizza CAN "rinfreshi" periodicamente l'informazione memorizzata all'interno della rete
- individuazione del fallimento di un nodo
  - ogni nodo invia periodicamente ai vicini messaggi di update contenenti le coordinate della sua zona ed una lista dei vicini con le proprie zone
  - l'assenza prolungata di messaggi di update da parte di un vicino viene interpretata come il fallimento del nodo
- se un nodo fallisce, la zona di cui esso è responsabile viene presa in carico da un nodo attivo sulla rete CAN.

# CAN: FALLIMENTO DI NODI

- Quando un nodo  $n$  individua il fallimento di un nodo vicino
  - inizializza un **timer** con un valore **proporzionale alla dimensione della sua zona**
  - quando il timeout scade, invia un messaggio di takeover ai vicini del nodo caduto
- I nodi che gestiscono zone più piccole inviano i loro messaggi per primi
- Quando un nodo riceve un messaggio di takeover, cancella il proprio timer, se la zona gestita ha dimensione inferiore a quella notificata nel messaggio
- In questo modo si sceglie come nodo di takeover **il nodo che gestisce la zona più piccola**
- Il nodo di takeover prende in consegna la zona del nodo fallito e la fonde con la sua, se possibile, altrimenti gestisce temporaneamente entrambe le zone