

Lezione n.9
Peer-to-Peer Systems
and Applications
Capitolo 8

CAN: Content Addressable Network
Laura Ricci

CAN: Content Addressable Network

- Idea Base: associa ad ogni nodo e ad ogni informazione un punto in uno spazio d -dimensionale
- Obiettivi
 - Scalabilità a centinaia di migliaia di nodi
 - Gestione efficace di inserimento e caduta di nodi
- Caratteristiche
 - 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

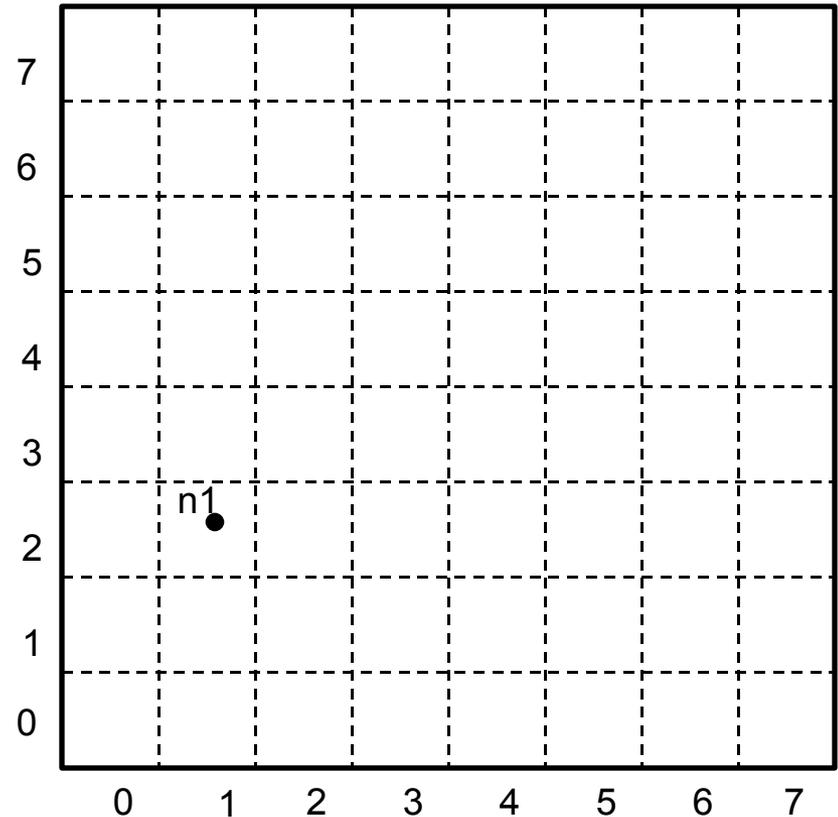
CAN: Content Addressable Network

- 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: Lo Spazio Bidimensionale

- Lo spazio è diviso tra i nodi
- L'intero spazio bidimensionale viene coperto dai nodi della rete CAN
- Ogni nodo copre un'area quadrata oppure un'area rettangolare in cui il rapporto tra base ed altezza è 1:2 oppure 2:1
- Esempio:

Il nodo $n1:(1, 2)$ è il primo nodo che entra nel sistema e copre l'intero spazio bi-dimensionale

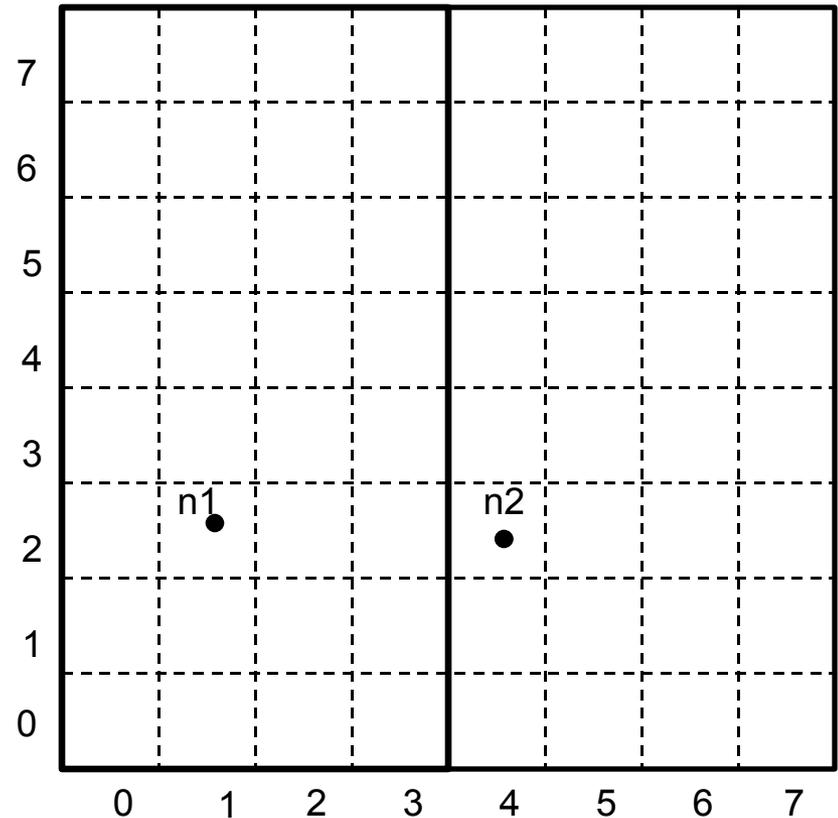


CAN: Lo Spazio Bidimensionale

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

⇒

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

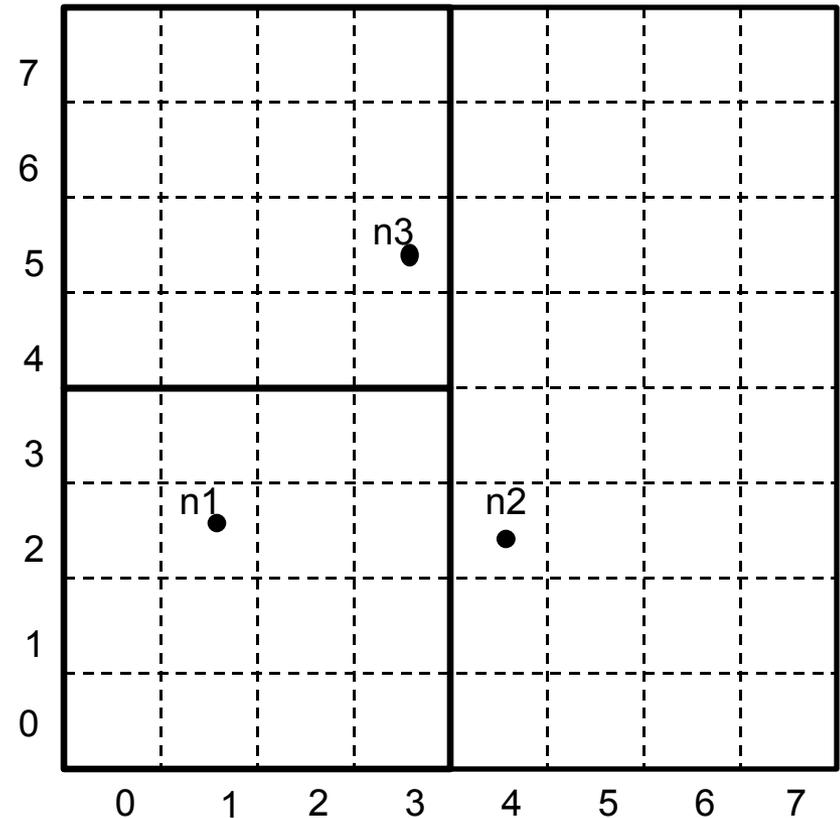


CAN: lo Spazio Bidimensionale

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

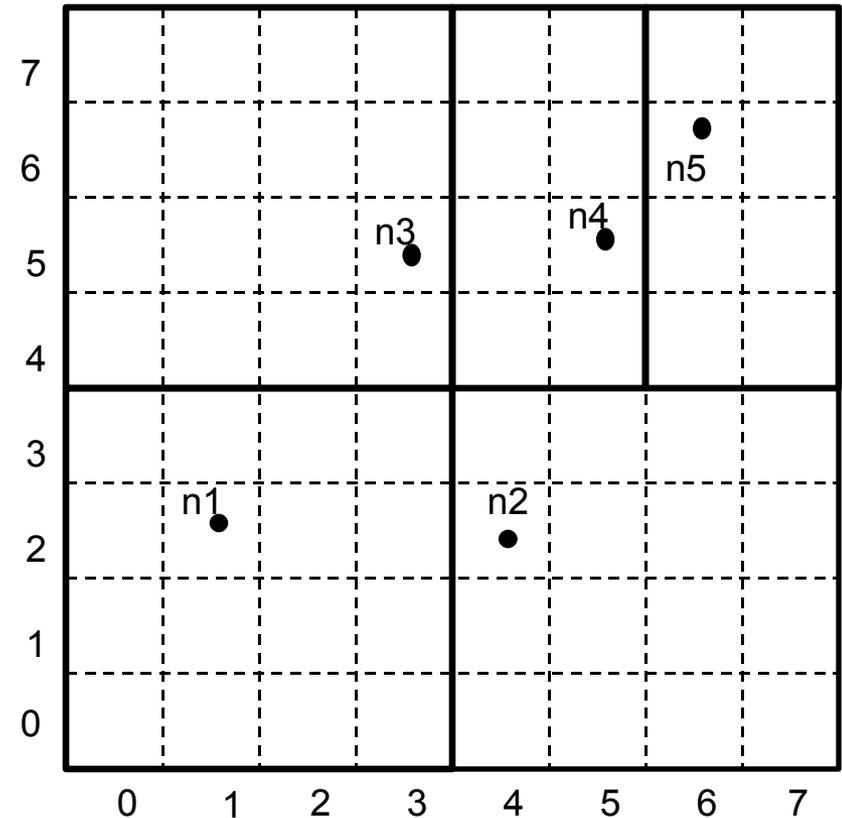
⇒

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



CAN: lo Spazio Bidimensionale

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



CAN: lo Spazio Bidimensionale

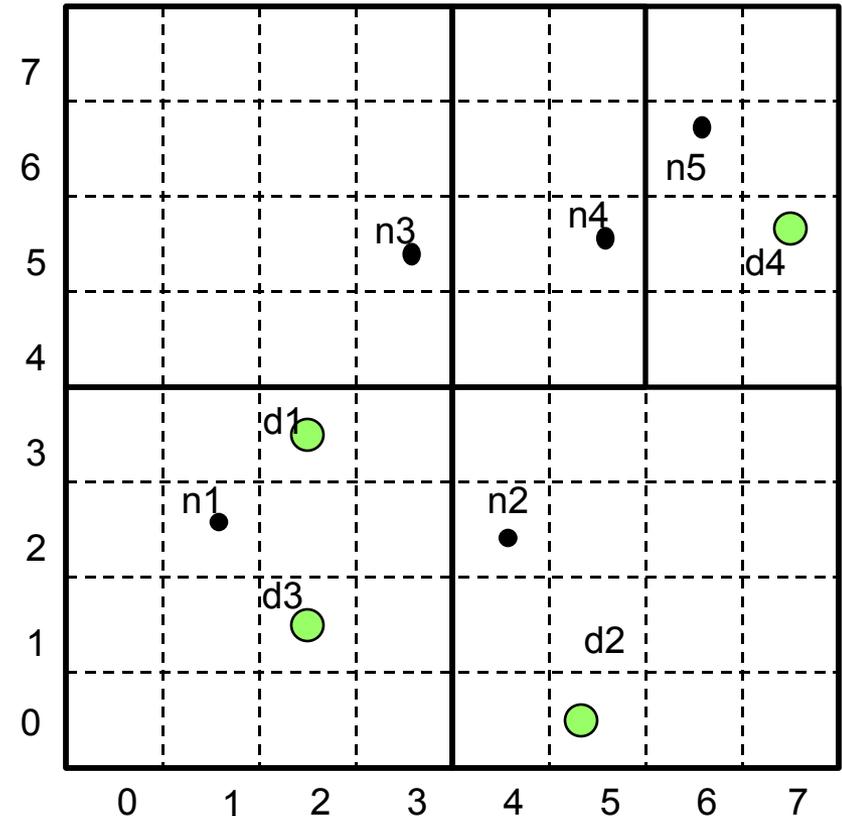
Nodi:

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

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

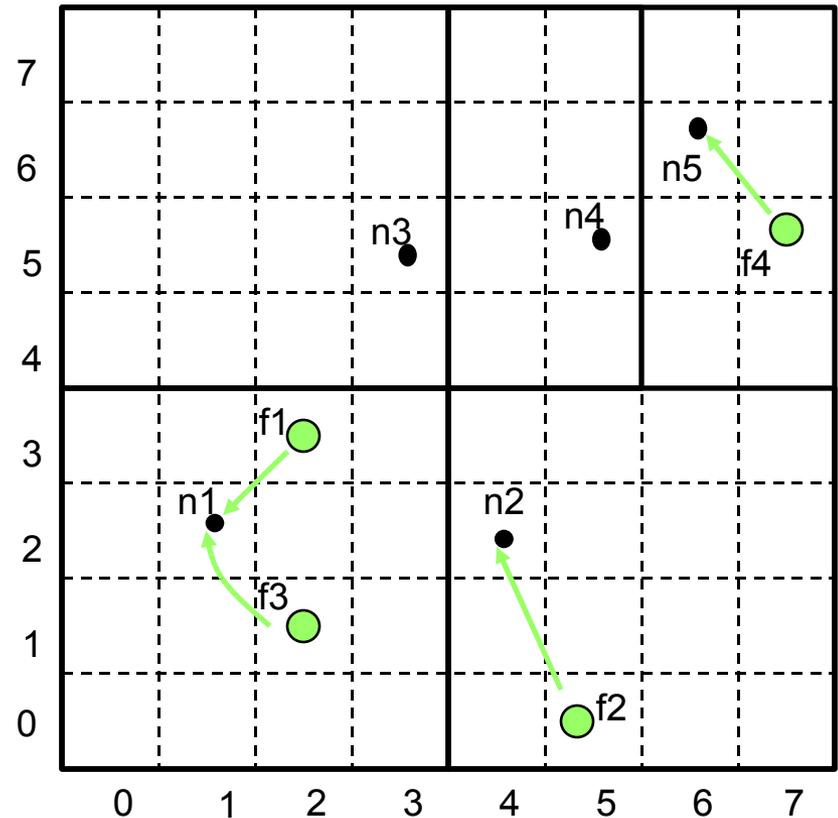
Dati

$d1:(2, 3)$; $d2:(5, 1)$; $d3:(2, 1)$; $d4:(7, 5)$;



CAN: Lo Spazio Bidimensionale

Ogni dato viene memorizzato dal nodo che gestisce la zona in cui è stato mappato



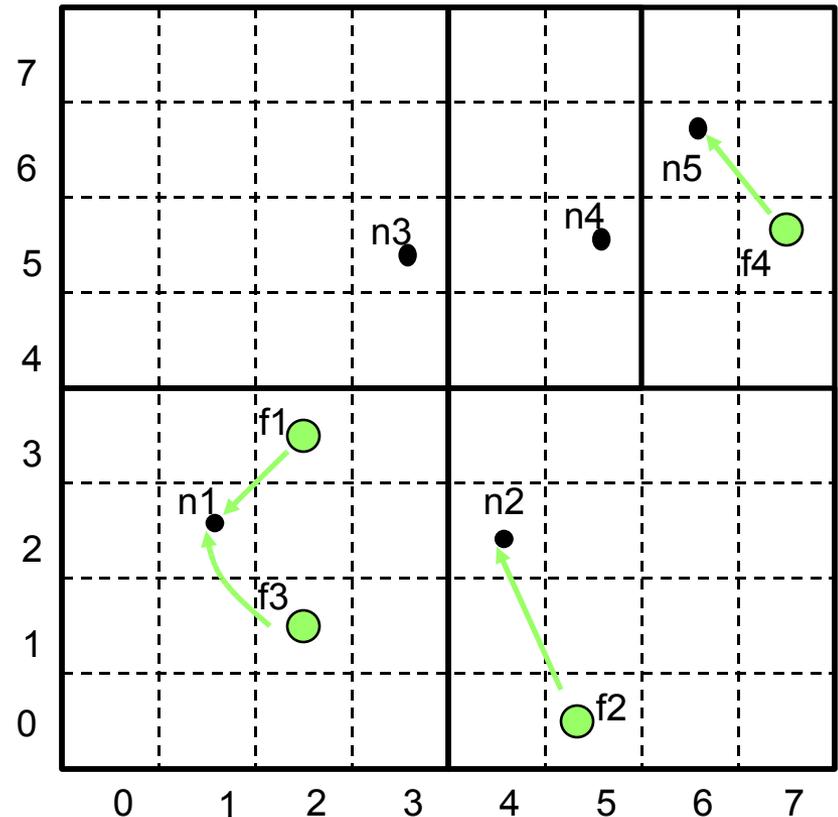
CAN: Ricerca dei Dati

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

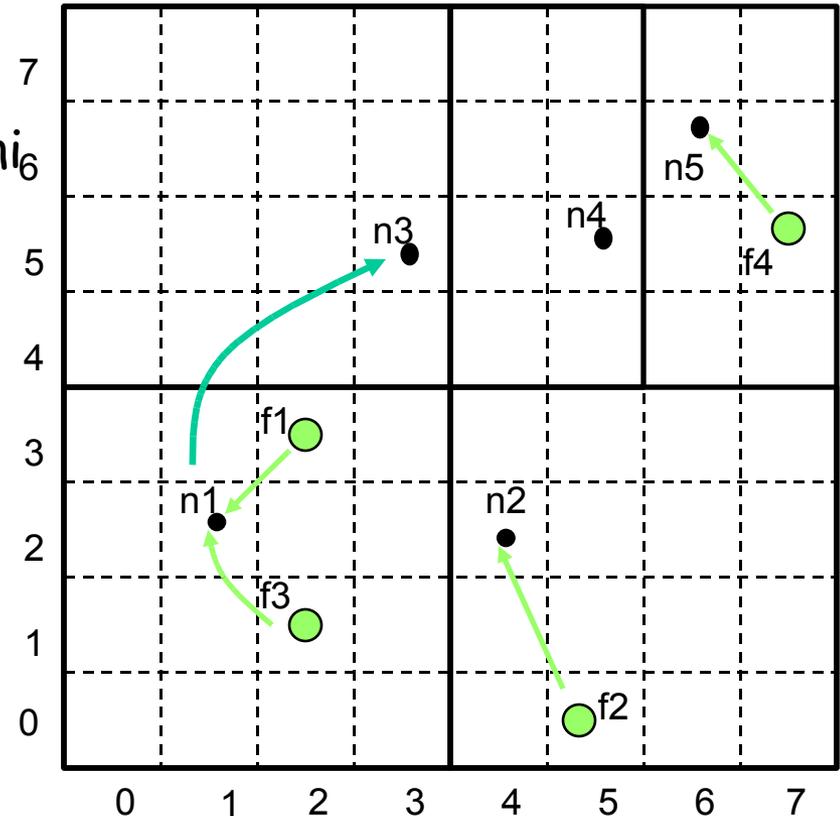
Esempio:

n1 ricerca f4



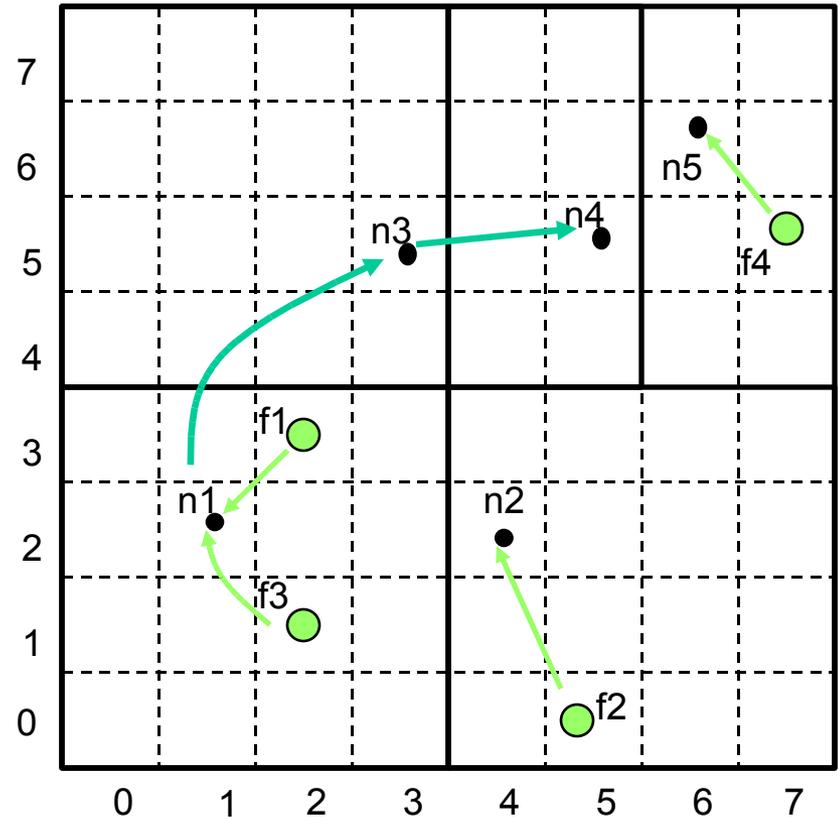
CAN: Ricerca dei Dati

- 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 f4



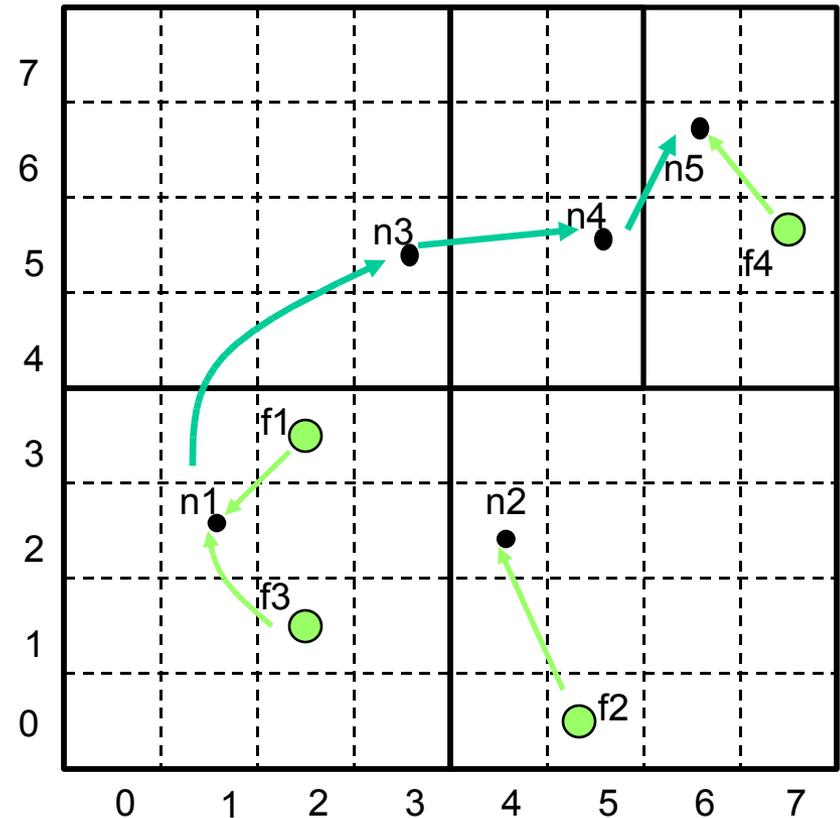
CAN: Ricerca dei Dati

- 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 f4



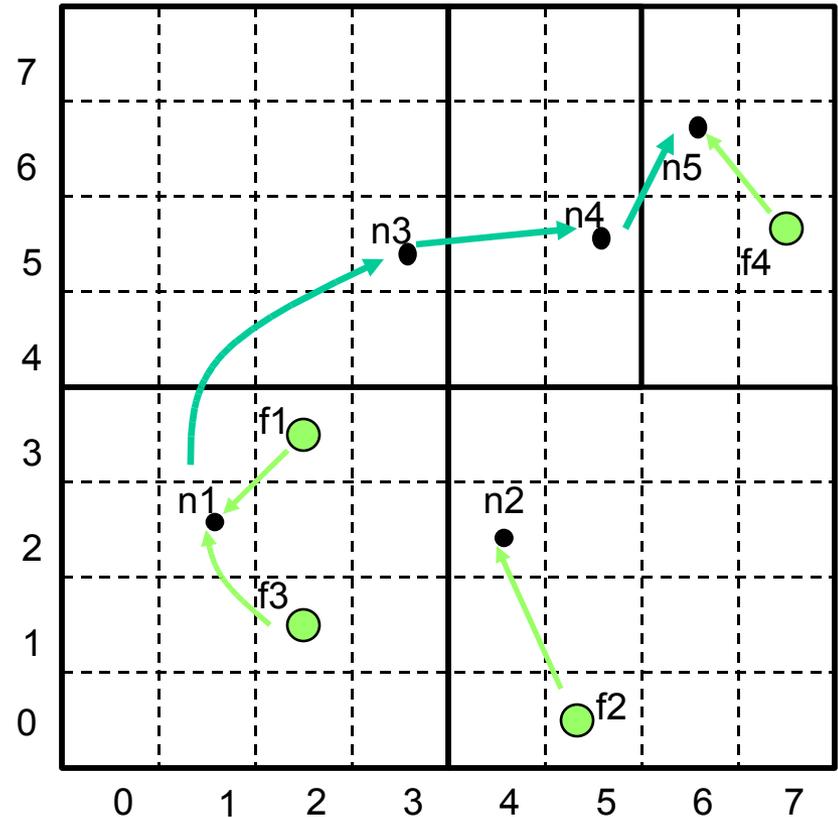
CAN: Ricerca dei Dati

- 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 f4



CAN: Ricerca dei Dati

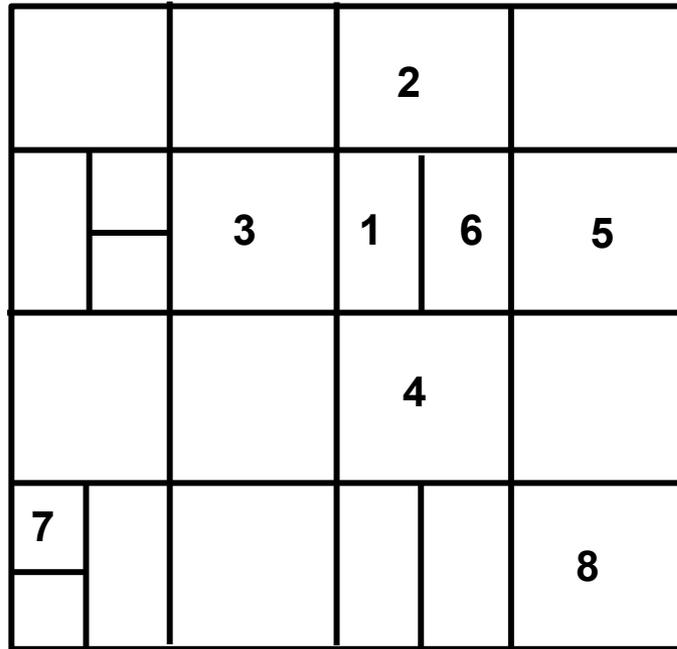
- Ogni nodo è a conoscenza dei suoi vicini nello spazio a d dimensioni
- La query viene inoltrata al nodo che risulta più vicino all'id della query
- Esempio:
n1 ricerca f4



CAN: Caratteristiche Generali

- 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"
- 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 si sovrappongono nella dimensione x e sono adiacenti rispetto alla dimensione $x-1$
- Lo spazio a D dimensioni è uno spazio logico e non presenta alcuna relazione con lo struttura della rete fisica sottostante

CAN: Caratteristiche Generali



- *esempio*: $D=2$
- vicini di 1: $\{2,3,4,6\}$
- vicini di 6: $\{1,2,4,5\}$
- considerare le richiuse: 7 ed 8 sono considerati vicini
- numero di vicini : $O(D) \Rightarrow$ al crescere del numero di nodi del sistema, l'informazione richiesta per il routing rimane proporzionale a D

Routing: Ogni nodo memorizza, per ogni vicino:

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

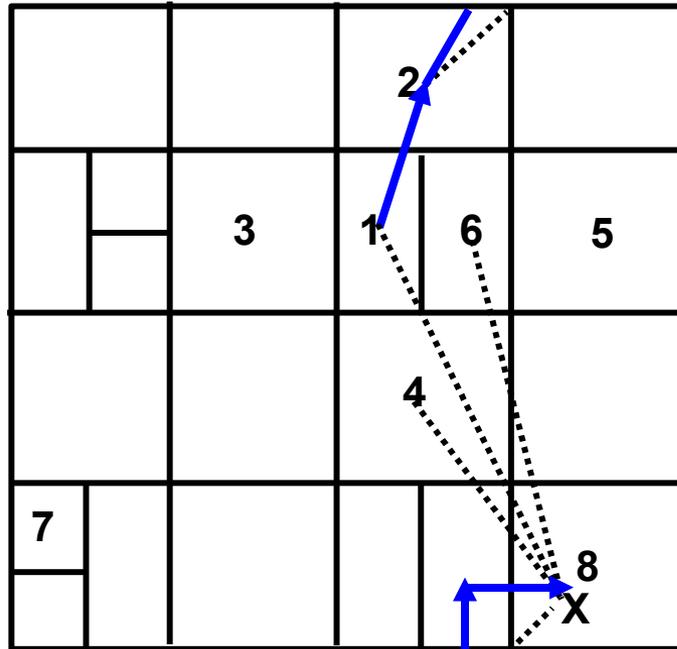
CAN: il Routing

- Ogni messaggio CAN contiene le coordinate di un punto P destinazione (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 .
- 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



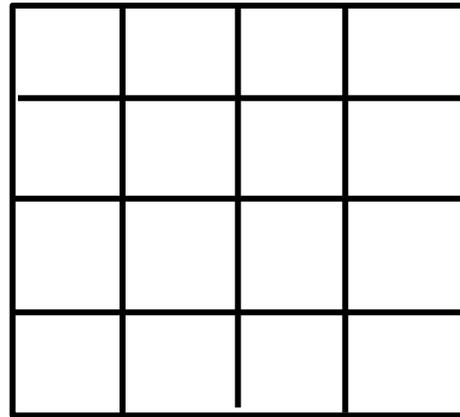
• *Esempio:* il nodo che gestisce la regione 1 deve inviare un messaggio al punto X

- 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
- numero medio di hops: $O(D \cdot N^{1/D})/4$

CAN: il Routing

Valutazione della complessità $O(D \cdot N^{1/D})/4$. Consideriamo uno spazio bi-dimensionale decomposto in quadrati



$N = 16$

- $N = 16$
- lato = $16^{\frac{1}{2}} = 4$
- al massimo devo fare $4/2$ hops per ogni dimensione (considerare le richiuse)
- numero massimo di hops per raggiungere un qualsiasi nodo $2 * 2 = 4$
- numero medio = 2

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

invia un messaggio di *JOIN*, utilizzando P come chiave. Il messaggio viene recapitato al nodo Z che gestisce la zona che contiene P

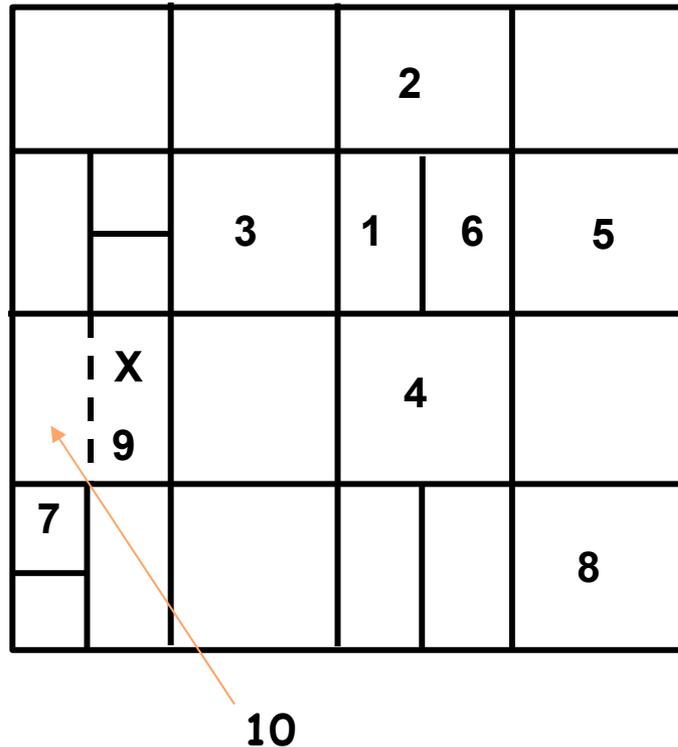
il messaggio di *JOIN* viene spedito mediante il nodo B

il messaggio raggiunge Z mediante il meccanismo di routing di CAN

CAN: Inserzione di Nuovi Nodi

- Il nodo n che riceve un messaggio di JOIN dal nodo z 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
- n tiene metà dello spazio per se e cede la restante parte a z
- n cede le chiavi mappate sullo spazio ceduto a z
- *Osservazione:* la probabilità che una zona venga scelta dipende dalla sua dimensione

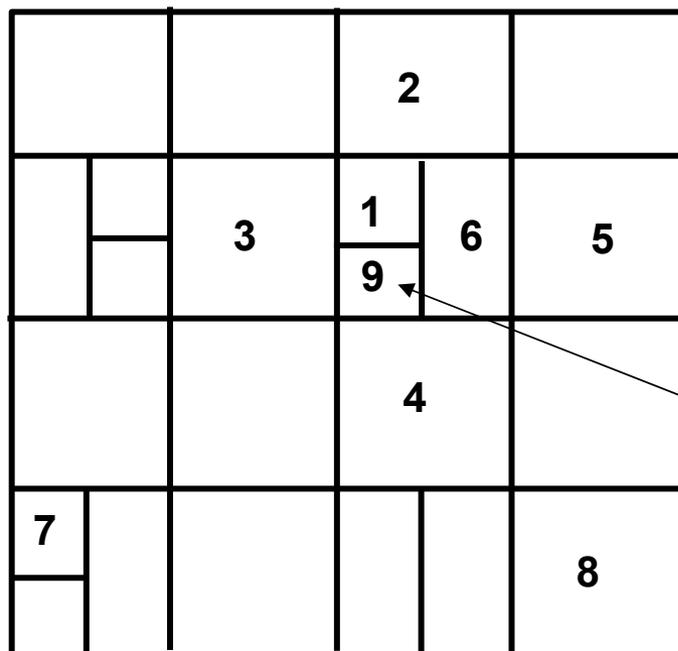
CAN: Inserzione di Nuovi Nodi



- viene generato l'identificatore del nuovo nodo (10)
- il nuovo nodo 10 sceglie, in modo casuale, il punto X
- X è contenuto nella zona gestita dal nodo 9
- 9 cede metà della propria zona a 10

CAN: Aggiornamento Informazioni di Routing

- Il nuovo nodo z riceve le informazioni relative ai suoi vicini dal nodo n che ha ceduto parte della propria zona
- i vicini di z sono un sottoinsieme dei vicini di n
- n 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 = {2,3,6,9}

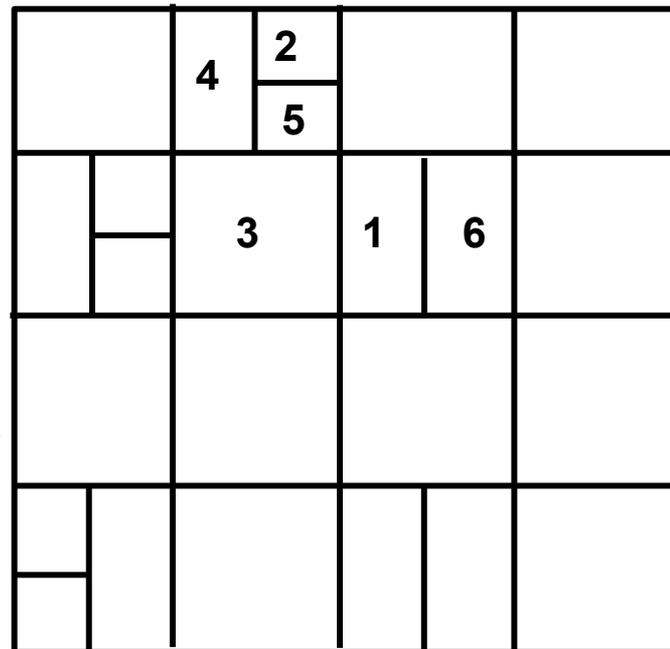
Nuovo nodo 9

CAN: Uscita dei Nodi dalla Rete

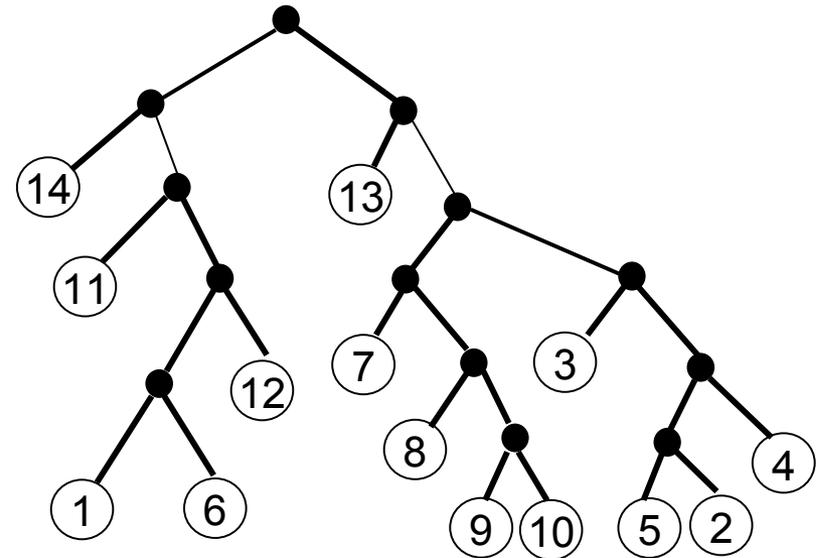
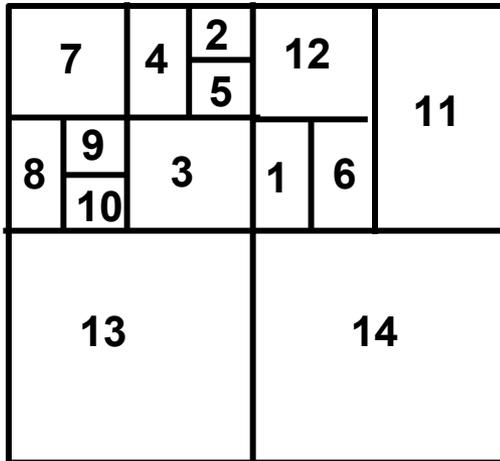
- Quando un nodo n decide volontariamente di uscire dalla rete, la gestione della zona z controllata da n deve essere passata ad un nodo vicino (takeover node)
- n consegna autonomamente le informazioni associate a z ad un nodo n' che controlla una zona vicina z' . Le informazioni trasmesse ad n' sono:
 - (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

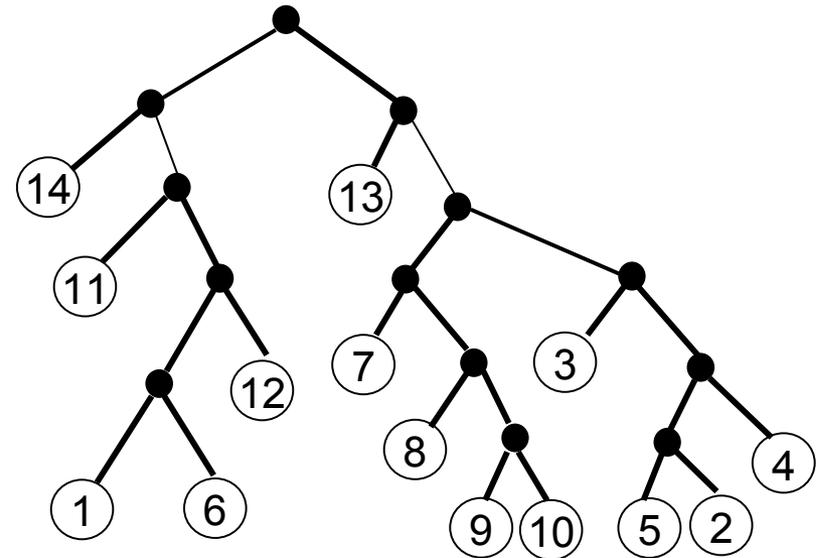


Il partizionamento dello spazio CAN puo' essere rappresentato mediante un **albero binario** in cui

- 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 TakeOver Node

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

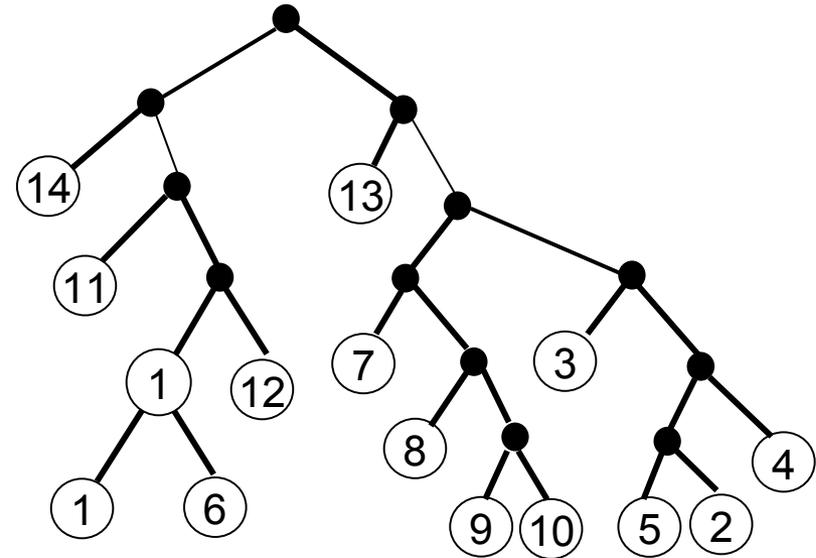


Caso 1: la foglia s viene rimossa, il fratello di s è una foglia t

- 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 TakeOver Node

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



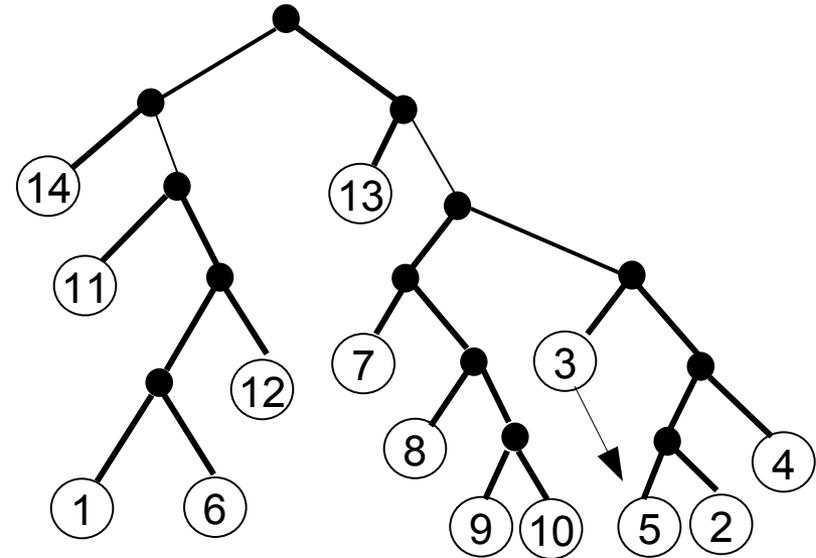
Caso 1: la foglia *s* viene rimossa, il fratello di *s* è una foglia *t*

Esempio: 6 lascia la rete CAN

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

CAN: Individuazione del Takeover Node

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

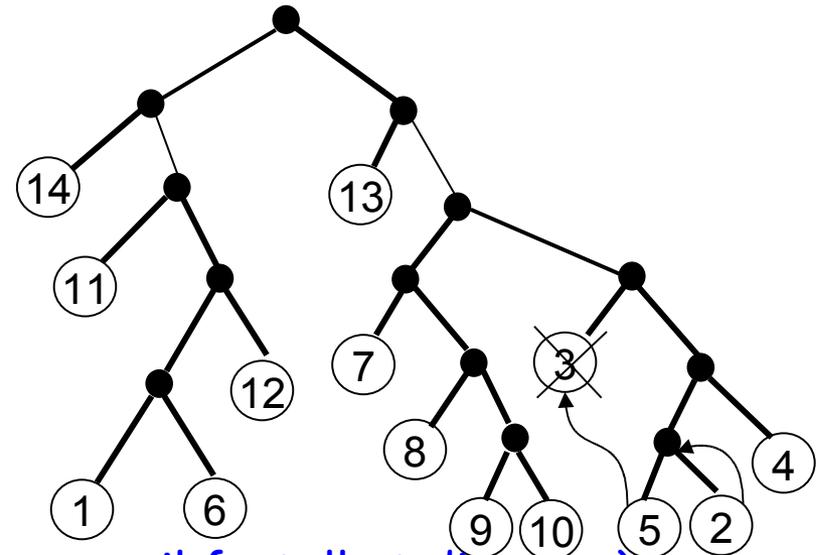


Caso 2: la foglia *s* viene rimossa, il fratello *t* di *s* non è una foglia

- La zona corrispondente viene presa in carico dal vicino che possiede la zona più piccola (esempio la zona 3 viene ceduta al nodo che gestisce la zona 5)
- Svantaggio: frammentazione delle zone

CAN: Individuazione del Takeover Node

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



Soluzione alternativa: la foglia *s* viene rimossa, il fratello *t* di *s* non è una foglia. Consente di evitare la frammentazione

- 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*.
- *z* prende in consegna la zona corrispondente ad *s*
- il fratello di *z* prende in consegna la zona gestita da *z*

CAN: Fallimento di Nodi

- CAN non prevede meccanismi di replicazione. Le informazioni memorizzate sul nodo fallito sono perse definitivamente
- l'applicazione che utilizza CAN deve "rinfrescare" 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