Lezione n.15

LA RETE EMULE-KADEMLIA

Peer-to-Peer Systems and Applications
Capitolo 8

KADEMLIA: CONCETTI GENERALI

- Protocollo P2P proposto da P. Maymounkov e D. Mazières (University of New York).

- Utilizza il protocollo UDP

- Le associazioni tra i clients ed i files che essi condividono
  - non viene registrata su server particolari

  - viene gestita dai clients presenti sulla rete Emule. Ogni client ha infatti anche la funzione di server.

- Ogni client agisce come server per l'associazione tra alcune chiavi e le rispettive fonti

- Obiettivo della ricerca: trovare quei client che hanno la responsabilità relativa alla chiave della ricerca in corso

KADEMLIA: CONCETTI GENERALI

- Sistema distribuito per la memorizzazione e la ricerca di coppie

  <key, value>
- Assegna mediante SHA un identificatore di 160 bits ai nodi ed alle chiavi
- Anche se presenta molte similitudini con altri sistemi (Chord, Pastry), presenta un insieme di caratteristiche non offerte simultaneamente da nessun sistema esistente
- Ogni coppia <key,value> viene assegnata al nodo il cui identificatore è più vicino alla chiave, secondo una certa nozione di vicinanza
- Metrica utilizzata per misurare la vicinanza/distanza tra gli identificatori:
  - XOR (OR Esclusivo) : definisce una metrica simmetrica (a differenze di sistemi tipo Chord)
  - In questo modo i clients Kademlia ricevono queries dagli stessi nodi contenuti nella propria routing table
  - Ogni nodo può arricchire la propria routing table mediante le informazioni contenute nelle queries

KADEMLIA:CONCETTI GENERALI

- Distanza tra due nodi (non geografica, ma sull'overlay) definita mediante l' OR ESCLUSIVO degli identificatori dei 2 nodi.

$$d(x, y) = (x \wedge \neg y) \vee (\neg x \wedge y)$$

$$d(x, x) = 0$$

$$d(x, y) > 0 \quad \text{se} \quad x \neq y$$

$$\forall x, y : d(x, y) = d(y, x)$$

$$d(x, y) + d(y, z) \geq d(x, z)$$

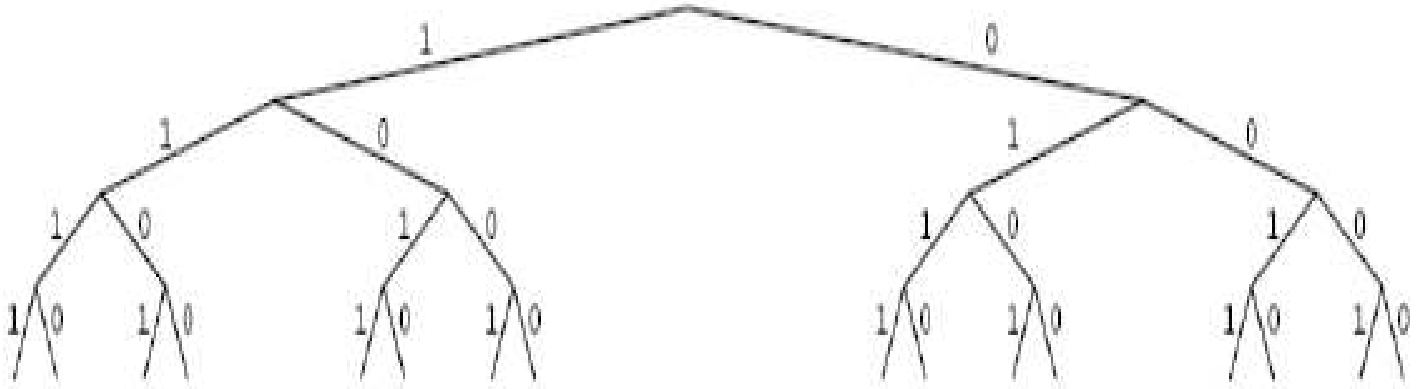
$$\exists y : d(x, z) = d(x, y) \text{ xor } d(y, z)$$

KADEMLIA: CONCETTI GENERALI

Algoritmo di look-up efficiente

- Lo stesso algoritmo utilizzato dall'inizio alla fine della ricerca (a differenza di Pastry...)
- Routing simile a Pastry: ogni passo approssimativamente dimezza la distanza con il target
- basato su queries parallele/asincrone
- Complessità O(log N)
- minimizza il numero di messaggi utilizzati dai nodi per acquisire conoscenza sul sistema

KADEMLIA: CONCETTI GENERALI



Distanza tra due nodi: proporzionale al numero di hops necessari, nell'albero per passare da un nodo all'altro

0111

0011

0111 XOR 0011= 0100 = 4

KADEMLIA: TABELLE DI ROUTING

- Tabelle di routing basate sulla costruzione di K-Buckets

- K-Bucket: Lista contenente (al massimo) K (K in genere = 20) contatti = triple del tipo

  <IP address, UDP port, Node ID>

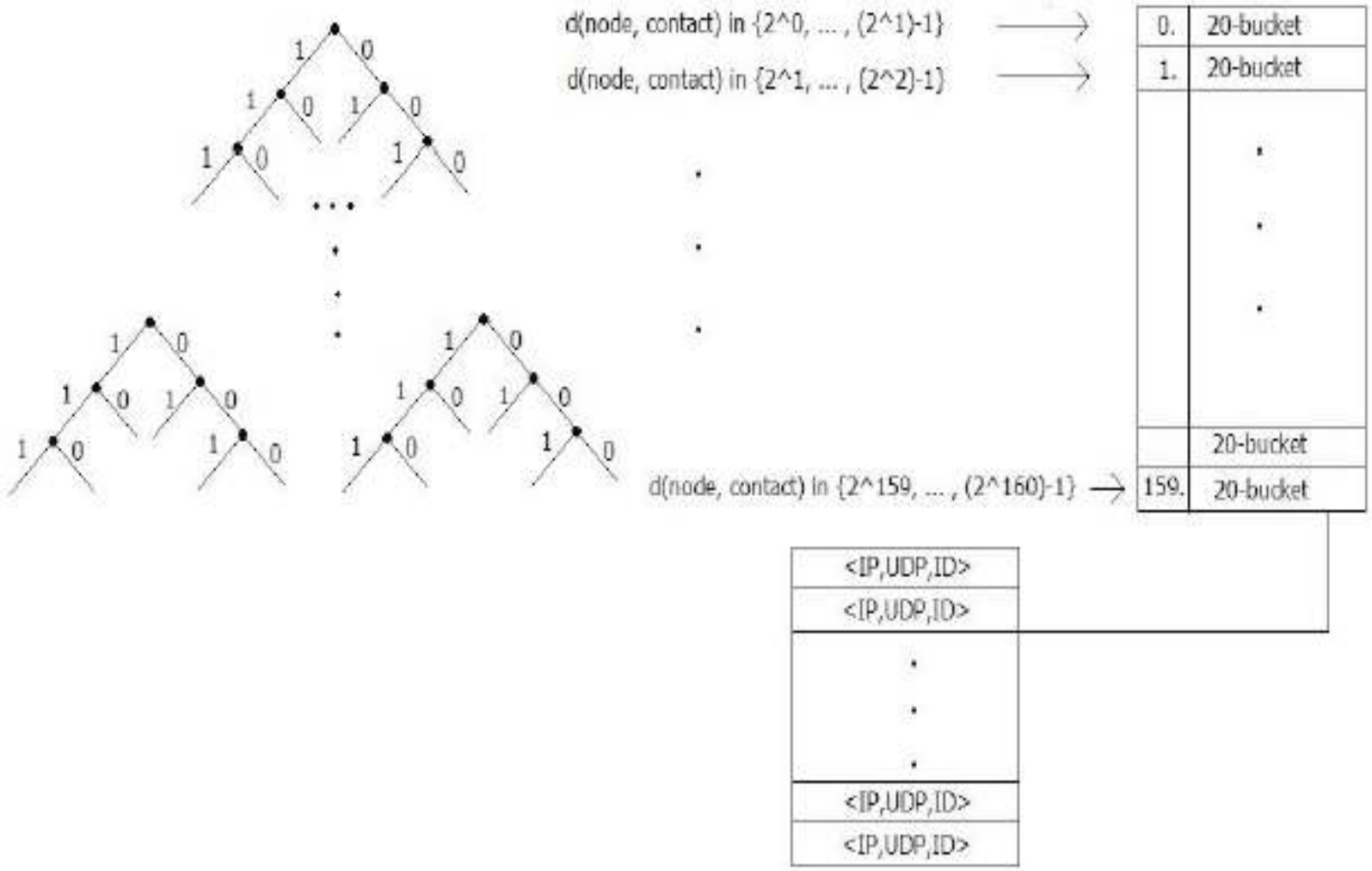
- La tabella di routing contiene B K-buckets, B =160 (lunghezza degli identificatori)

- Ogni nodo tiene una lista di contatti di nodi che si trovano ad una distanza compresa tra 2^i e 2^{i+1} da quel nodo, per ogni $0 \leq i \leq 160$

- Dato il K-bucket X di indice j presente nalla tabella di routing del nodo node e considerato un contatto contact nel bucket X, vale che

$$\forall\, 0 \leq j < B : 2^j \leq d(node, contact) < 2^{j+1}$$

KADEMLIA: LE TABELLE DI ROUTING

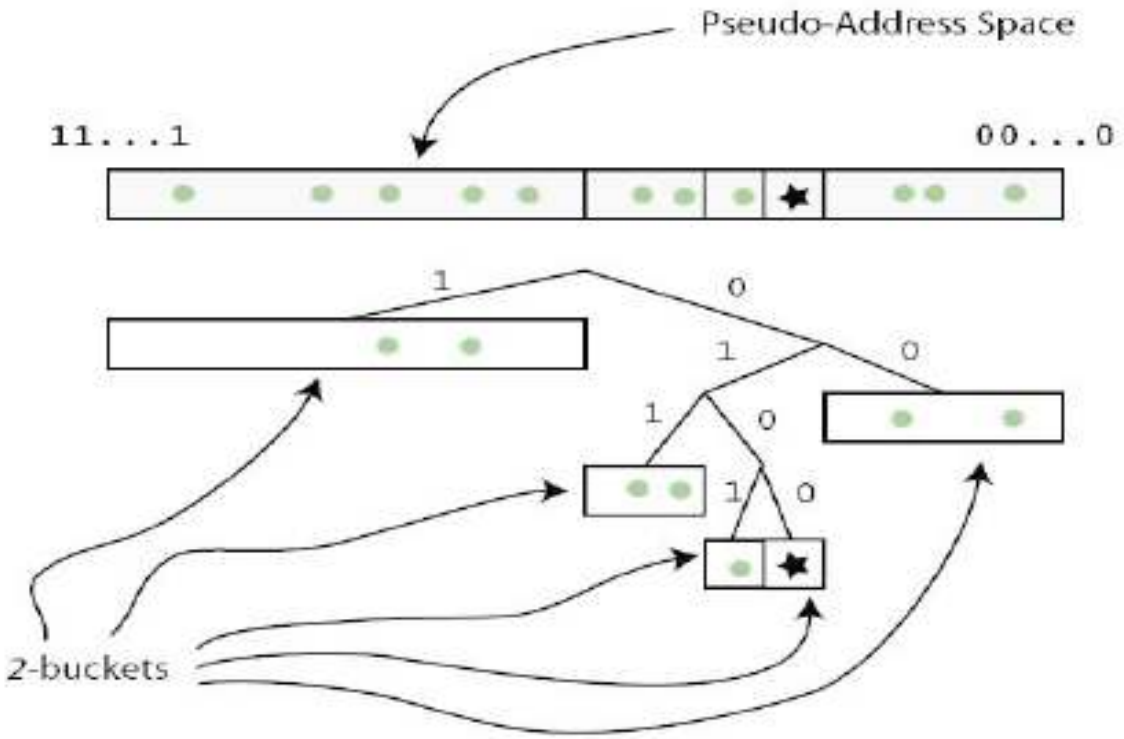


KADEMLIA: LA TABELLA DI ROUTING

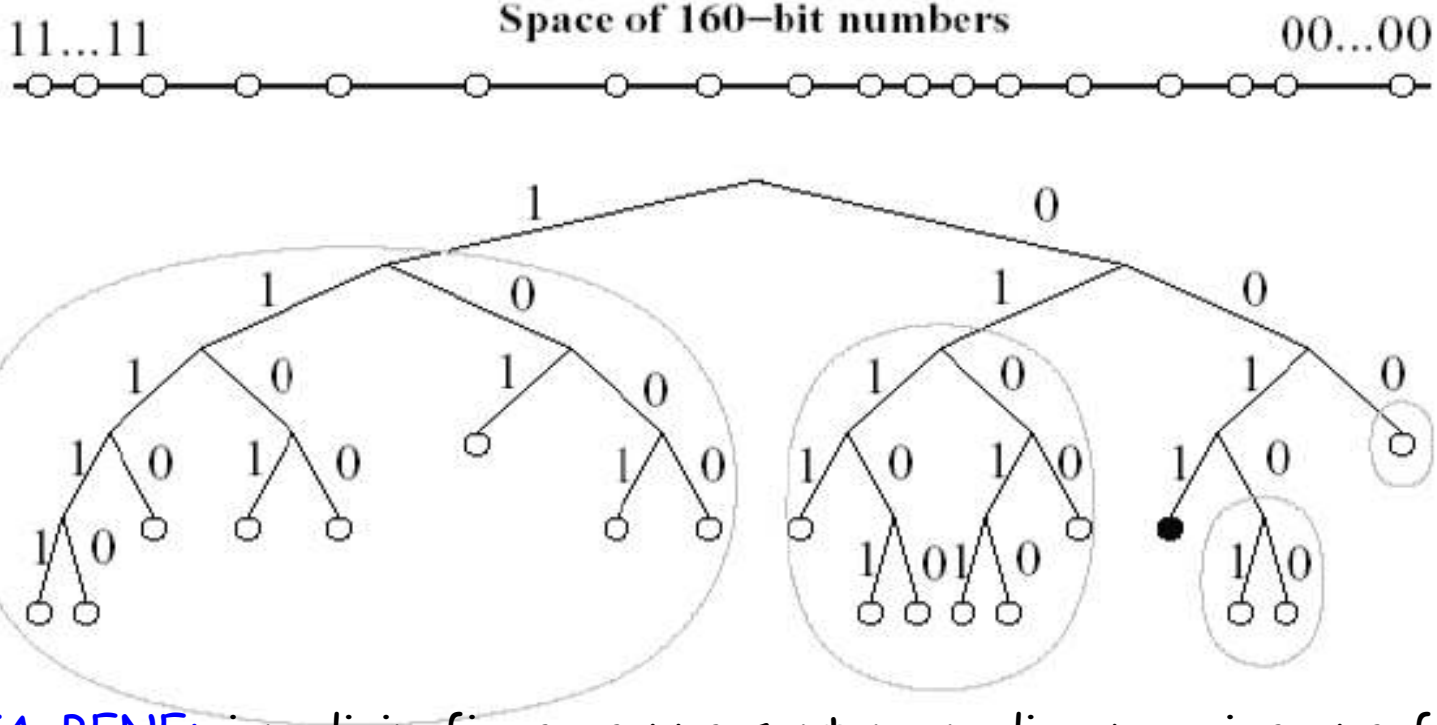
- Le prime entrate della tabella di routing (valori piccoli di j) contengono poche entrate
- Per valori alti di j ogni bucket contiene un maggior numero di contatti, ma mai più di K
- Ogni K bucket contiene i nodi che presentano un certo prefisso comune con il nodo dato
- Ogni K bucket copre un sottospazio dello spazio degli identificatori
- Insieme tutti i k-buckets coprono l'intero spazio degli identificatori

KADEMLIA: LE TABELLE DI ROUTING

- La tabella di routing è un albero binario, dove le foglie sono i k buckets



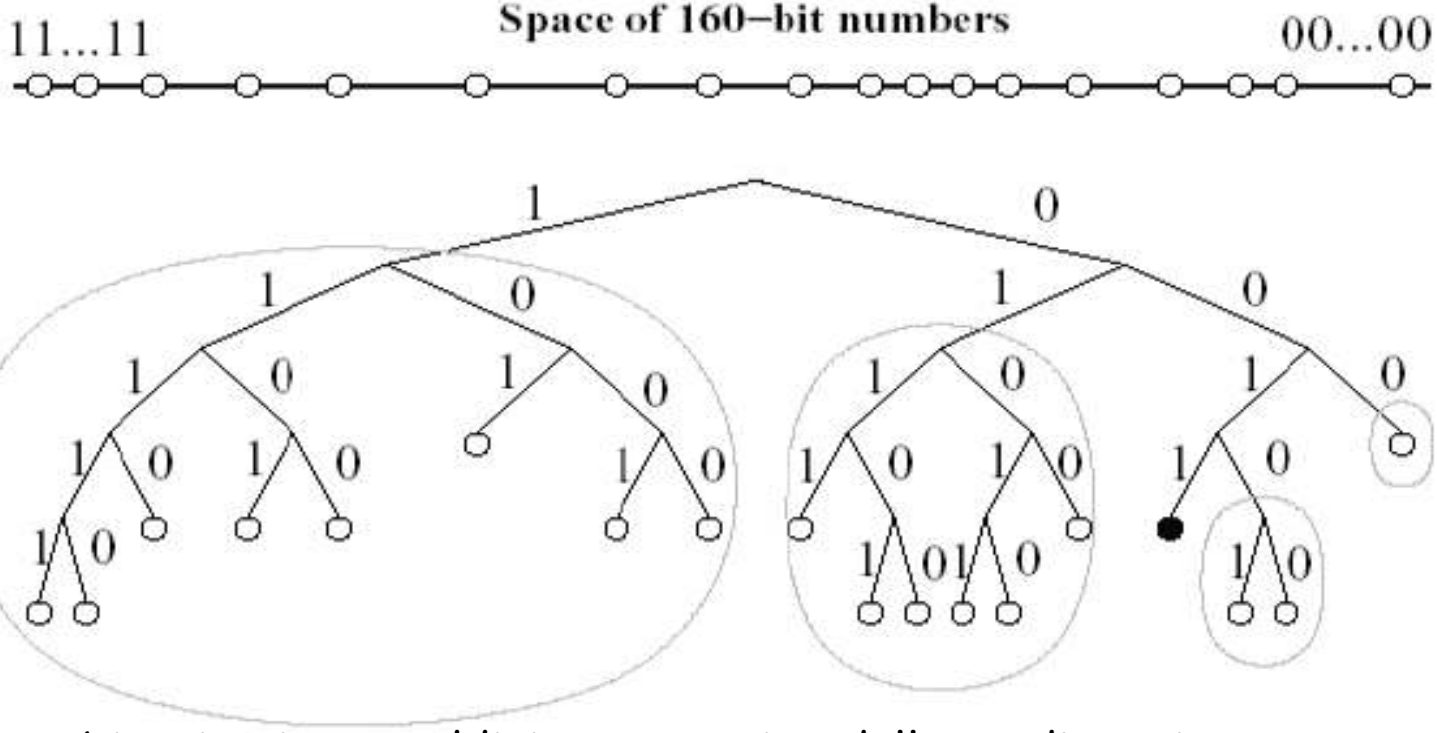
KADEMLIA:LE TABELLE DI ROUTING



• NOTA BENE: i nodi in figura rappresentano nodi a maggiore profondità nell'albero che sono univocamente individuati dal prefisso evidenziato

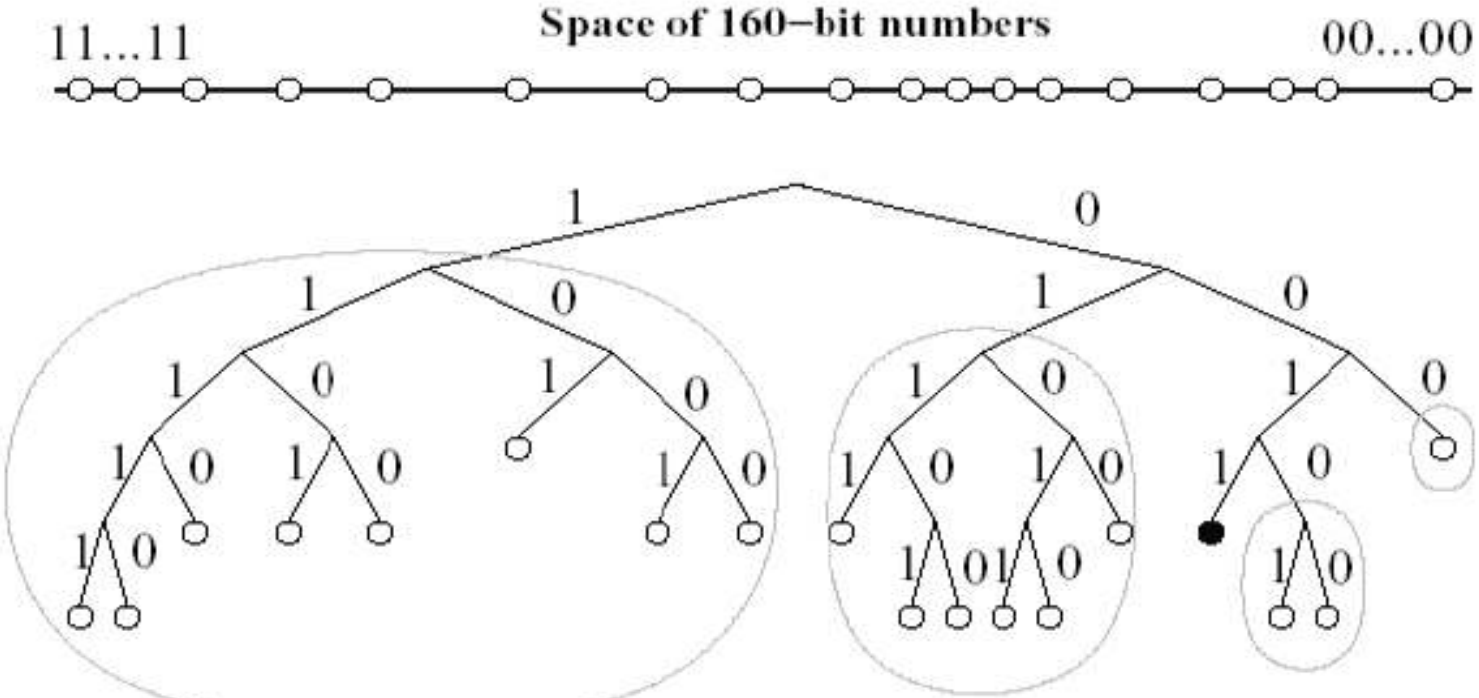
•consideriamo un nodo n, ad esempio il nodo nero evidenziato in figura identifica univocamente il nodo con prefisso 0011. Dallo spazio degli identificatori si deduce che c'è un solo nodo con quel prefisso

KADEMLIA:LE TABELLE DI ROUTING



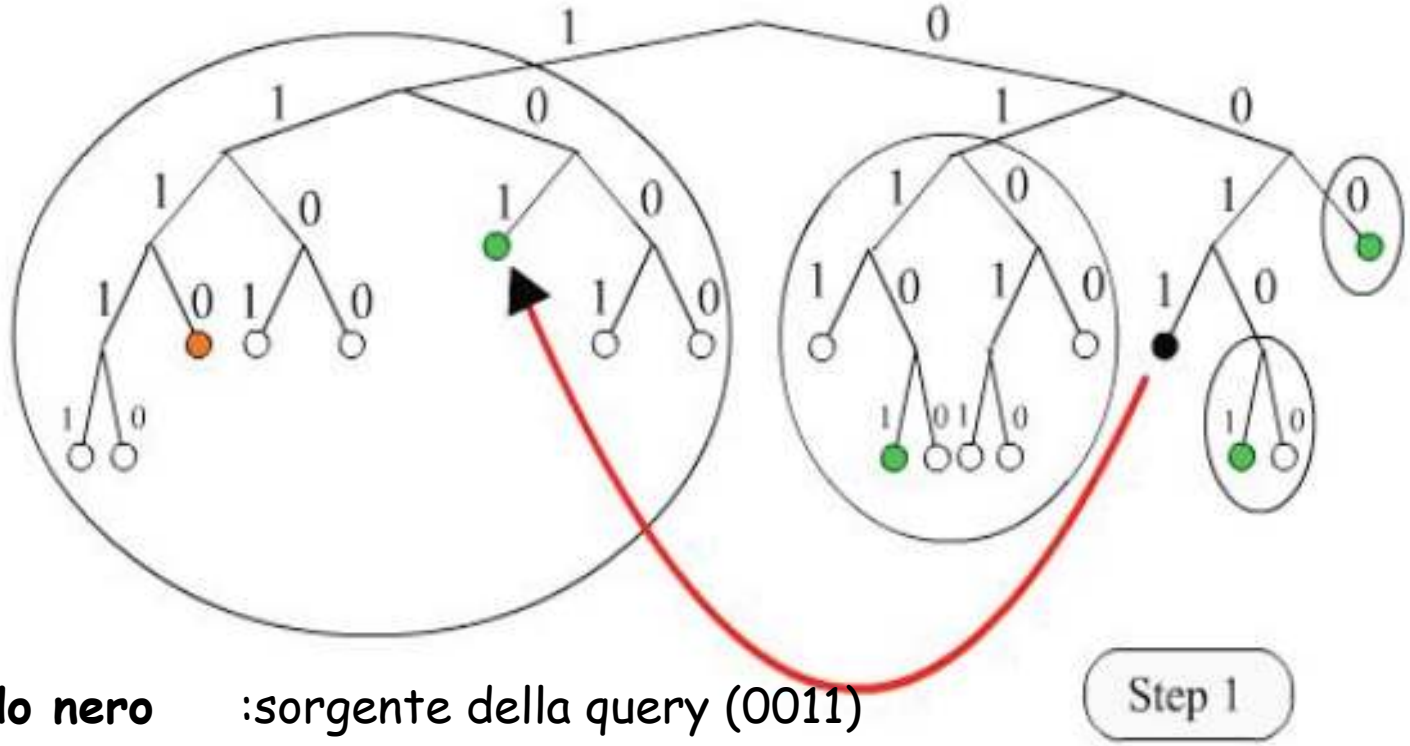
- l'albero binario viene suddiviso, a partire dalla radice, in una successione di sottoalberi sempre più piccoli che non contengono n
- il primo sottoalbero contiene metà dell'albero binario che non contiene n
- Il secondo sottoalbero contiene la metà della rimanente parte dell'albero che non contiene n e così via

KADEMLIA: LE TABELLE DI ROUTING



- Le tabelle di routing di Kademlia assicurano che ogni nodo conosca alcuni nodi in ogni sottoalbero così individuato
- Ogni query viene inviata ad un nodo in un sottoalbero
- Il nodo contattato propaga la query verso sottoalberi via via più vicini al target

KADEMLIA: IL ROUTING

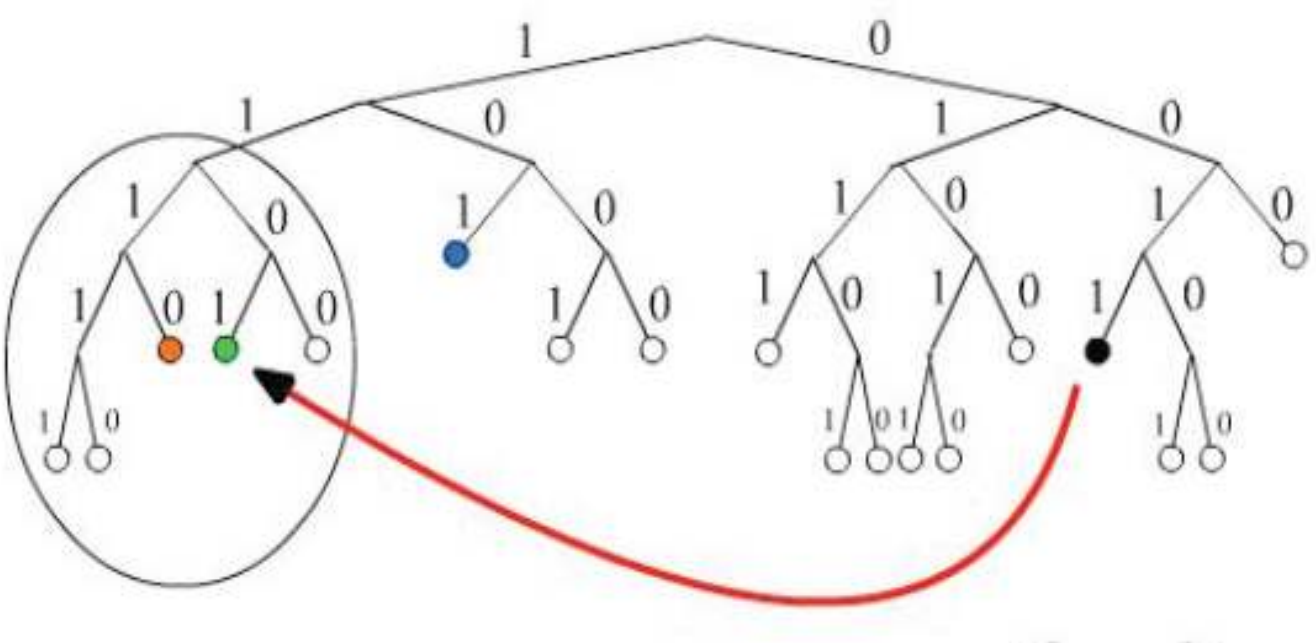


Nodo nero :sorgente della query (0011)

Nodo arancio : target della query (1110)

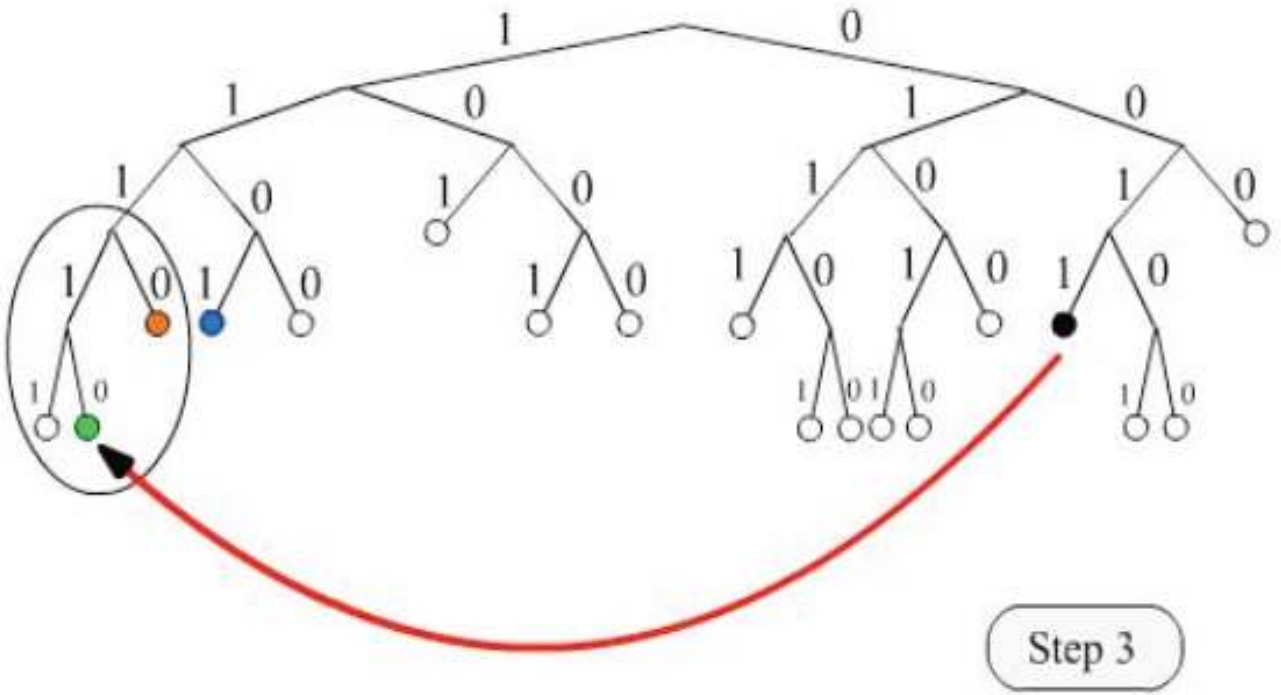
Nodi verde : nodi conosciuti dalla sorgente negli altri sottoalberi

KADMELIA: IL ROUTING



Nodo nero : sorgente della query (0011)
Nodo arancio : target della query (1110)
Nodi verde : nodi conosciuti dalla sorgente negli altri sottoalberi
Nodo Blu : restituisce alla sorgente un ulteriore nodo da contattare

KADMELIA: IL ROUTING



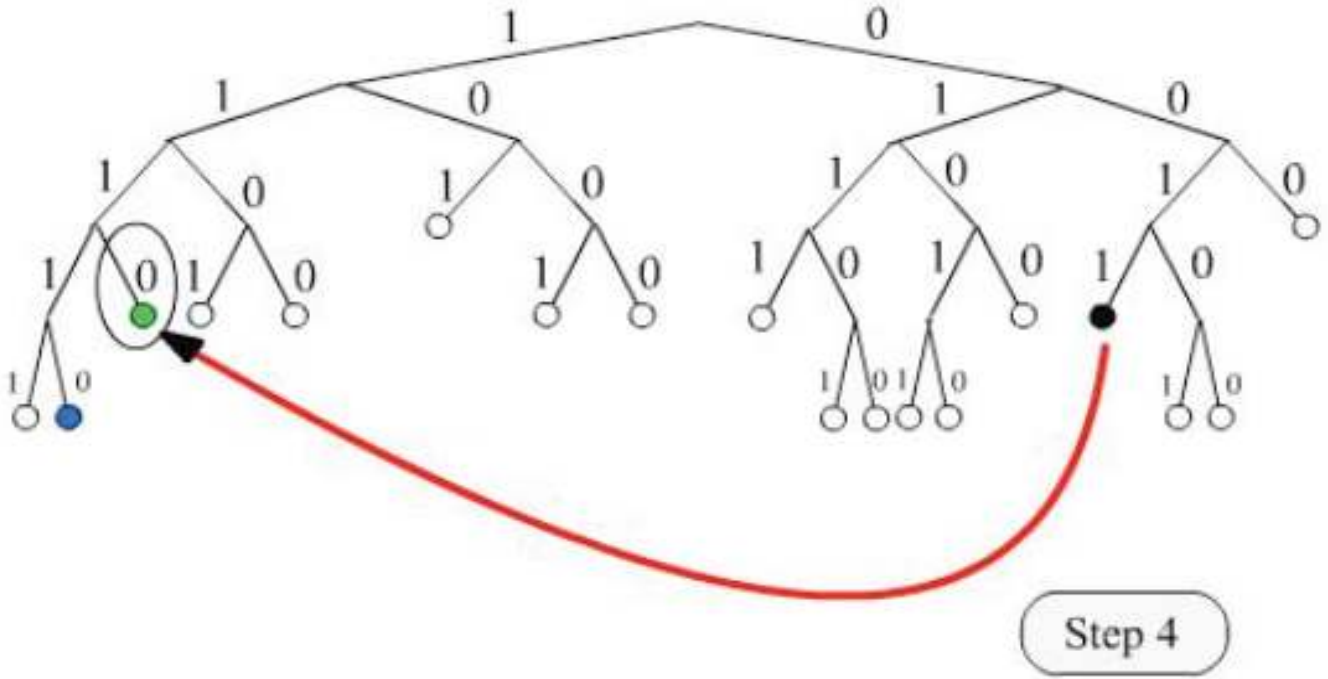
Nodo nero : sorgente della query (0011)

Nodo arancio : target della query(1110)

Nodi verde : nodi conosciuti dalla sorgente negli altri sottoalberi

Nodo Blu : restituisce alla sorgente un ulteriore nodo da contattare

KADMELIA: IL ROUTING



Nodo nero : sorgente della query (0011)
Nodo arancio : target della query (1110)
Nodi verde : nodi rappresentanti negli altri sottoalberi
Nodo Blu : restituisce alla sorgente un ulteriore nodo da contattare

KADEMLIA: IL ROUTING