Lezione n.4
DISTRIBUTED HASH TABLES: INTRODUZIONE

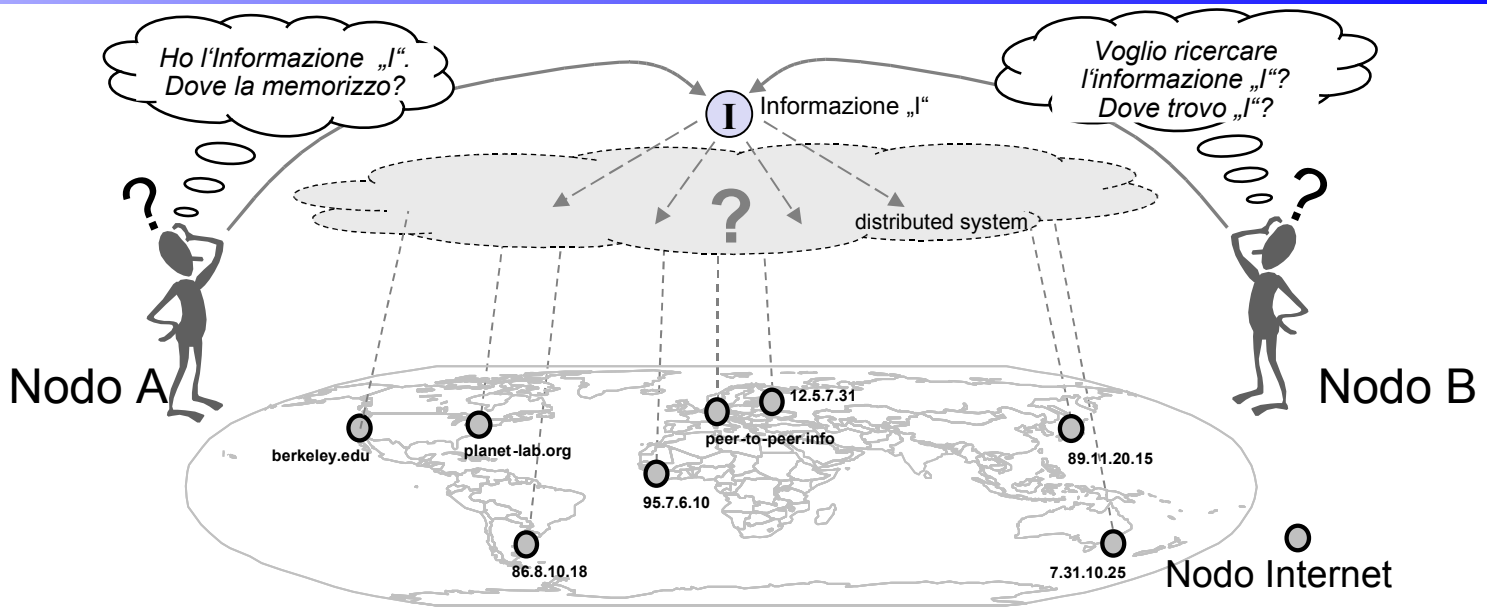
6/3/2009

Laura Ricci

DISTRIBUTED HASH TABLES:INTRODUZIONE

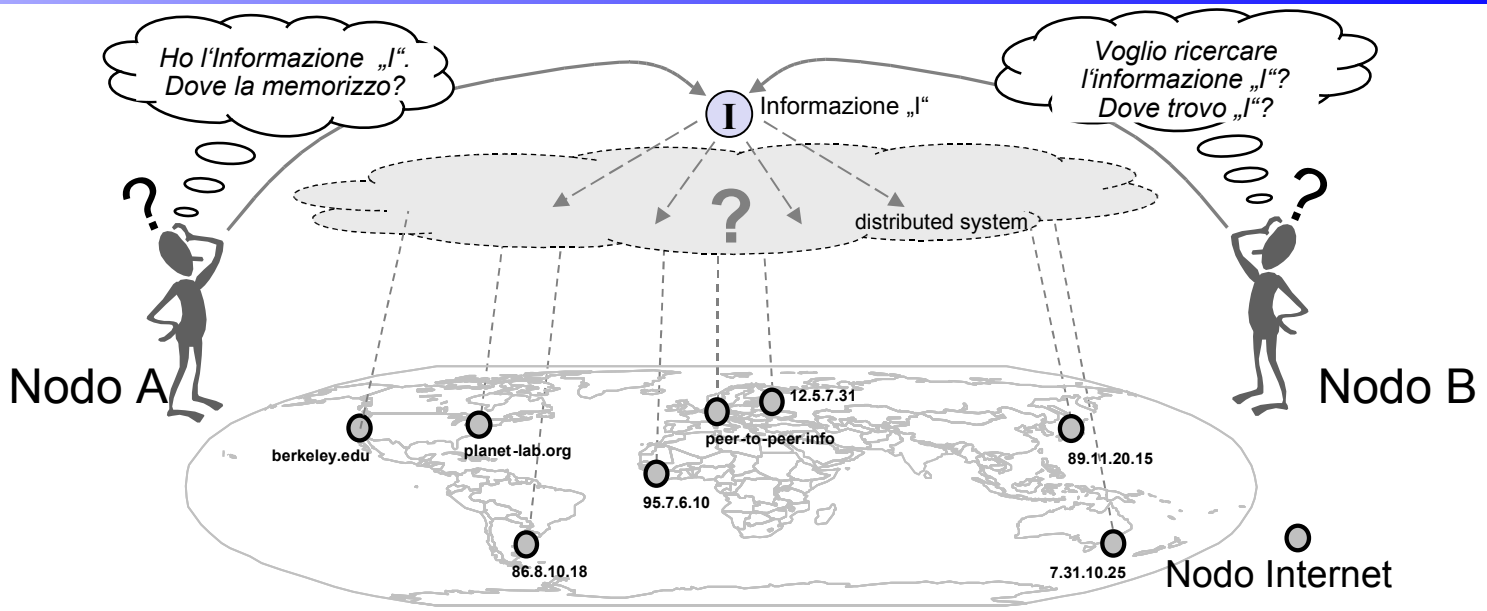
- Distributed Hash Tables (DHT): Introduzione
  - Motivazioni
  - Caratteristiche
  - Confronti
- DHT: Aspetti Fondamentali
  - Gestione distribuita dei dati
  - Indirizzamento nelle Distributed Hash Tables
  - Routing
  - Memorizzazione dei dati
- DHT: I meccanismi
  - Inserzione di nuovi nodi
  - Fallimento/Uscita volontaria di nodi dal sistema
- DHT: Le interfacce
- Conclusioni

SISTEMI P2P: IL PROBLEMA DELLA RICERCA



- Nei sistemi non strutturati analizzati nelle lezioni precedenti le risorse messe a disposizione di un peer sono memorizzate dal peer stesso
- Il problema principale, dovuto alla mancanza di strutturazione della rete è quello della ricerca. Dove si trova l'informazione con le caratteristiche desiderate?
- I sistemi strutturati sono basati su assunzioni diversa:
  - Una informazione condivisa può essere memorizzata su qualsiasi nodo della rete
  - La associazione delle informazioni ai nodi avviene secondo un criterio ben preciso che consente poi un routing 'intelligente' delle query verso I nodi che possono soddisfarle

SISTEMI P2P: IL PROBLEMA DELLA RICERCA



- A memorizza una informazione I all'interno del sistema distribuito, B vuole reperire I,ma non conoscere a priori l'effettiva locazione di I
- Come organizzare il sistema distribuito? In particolar modo, quali sono i meccanismi utilizzati per decidere dove memorizzare l'informazione e come reperirla?
- Qualsiasi soluzione deve tenere particolarmente in considerazione
  - Scalabilità del Sistema. Occorre controllare l'overhead di comunicazione e la memoria utilizzata da ogni nodo, in funzione del numero dei nodi del sistema
  - Robustezza ed adattabilità in caso di faults e frequenti cambiamenti

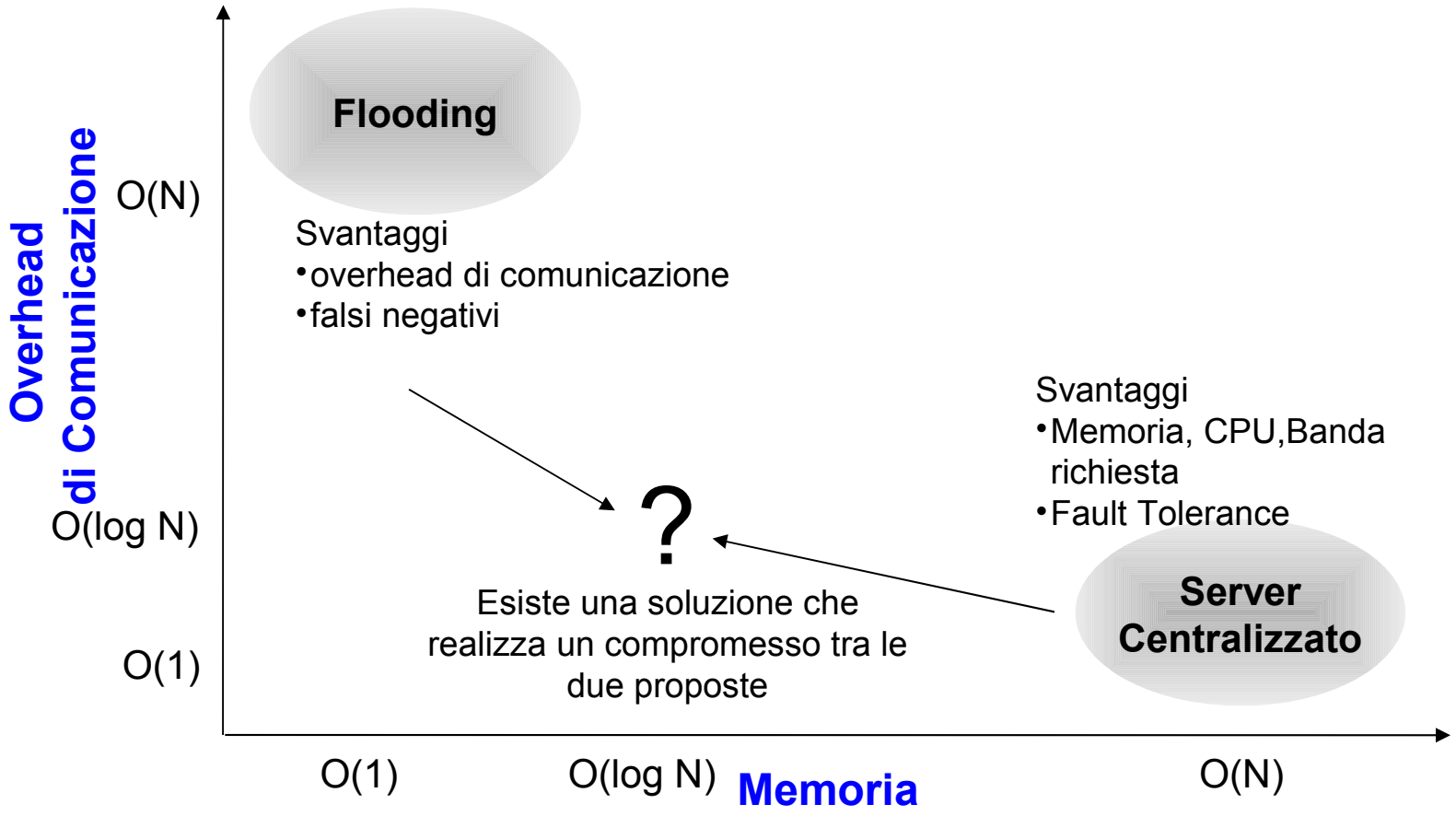
P2P:ANALISI e CONFRONTI

- Approccio Centralizzato
  - Ricerca: O(1) – "memorizzo l'informazione su un server centralizzato"
  - Quantità di Memoria Richiesta sul Server: O(N) (N = numero di informazioni disponibili nel sistema)
  - Banda richiesta (connessione server/rete): O(N)
  - Possibilità di sottomettere al sistema queries complesse

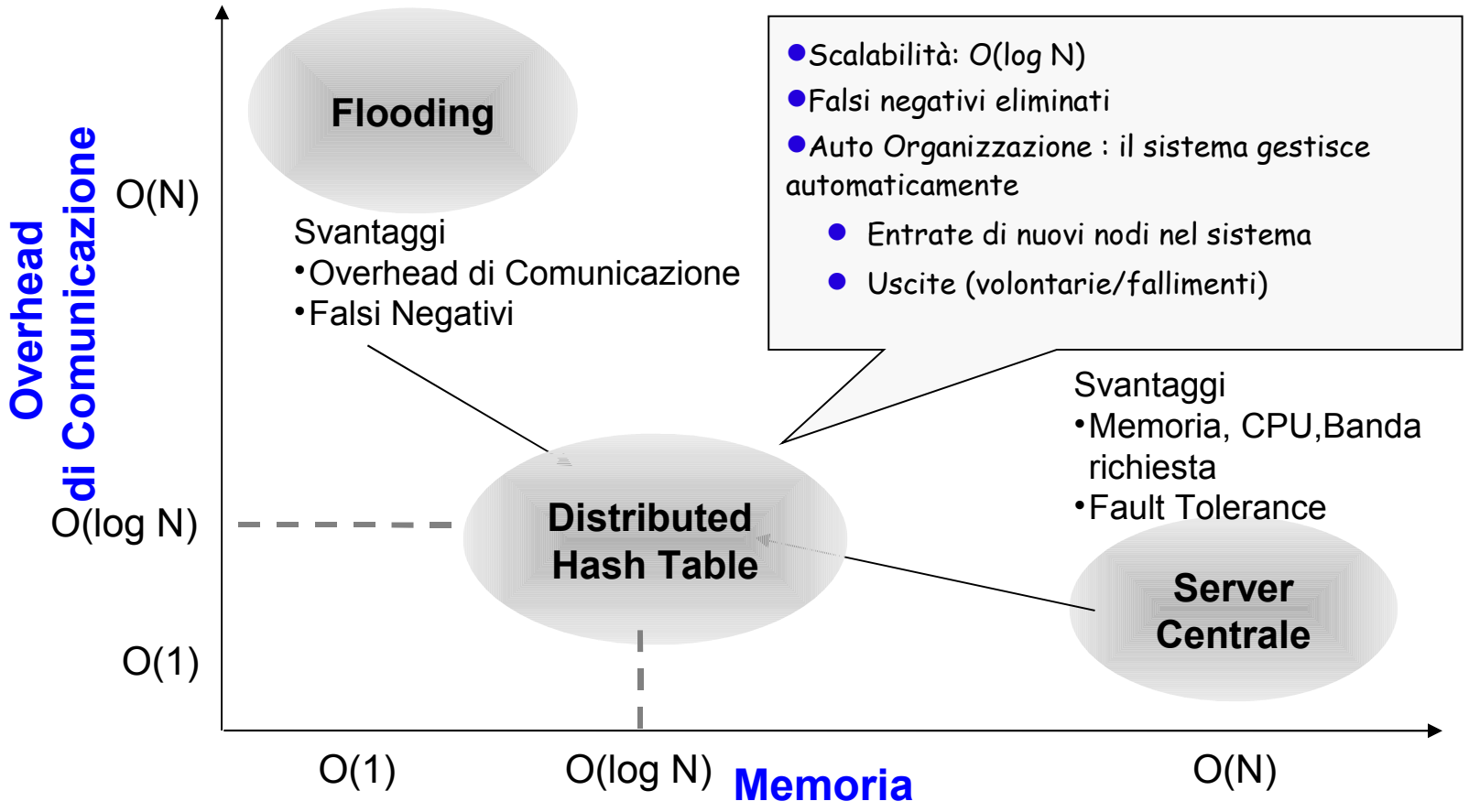
- Approccio Completamente Distribuito
  - Ricerca: caso pessimo $O(N^2)$ - "ogni nodo chiede a tutti i vicini ". Possibili ottimizzazioni (TTL, identificatori per evitare cammini ciclici)
  - Quantità di memoria richiesta : O(1)
    - Informazione condivisa: non dipende dal numero di nodi del sistema
    - Non si utilizzano strutture dati per ottimizzare il routing della query (flooding)

DISTRIBUTED HASH TABLES:MOTIVAZIONI

Analisi dei sistemi Esistenti



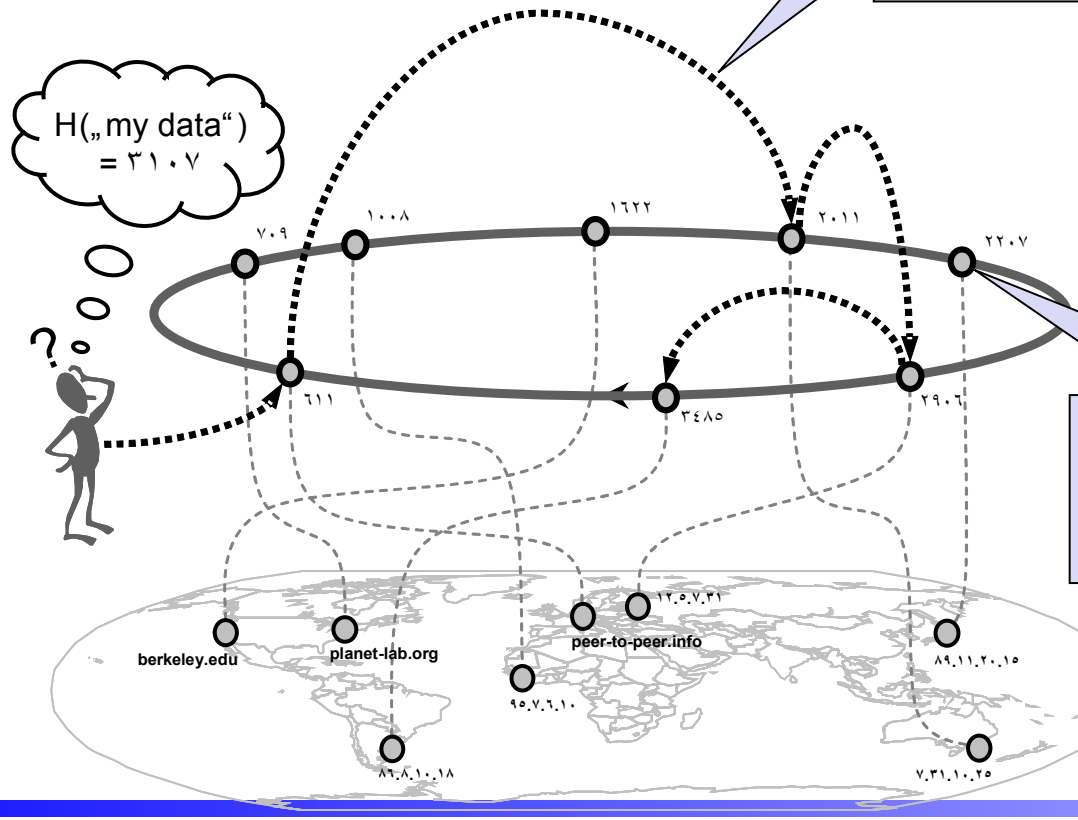
DISTRIBUTED HASH TABLES: MOTIVAZIONI



DISTRIBUTED HASH TABLES: CONCETTI GENERALI

- Obiettivi
  - O(log(N)) hops per la ricerca di un'informazione
  - O(log(N)) entrate nella tabella di routing

Il routing richiede O(*log(N)*) passi per raggiungere il nodo che memorizza l'informazione



O(*log(N)*) dimensione della tabella di routing di ogni nodo

DISTRIBUTED HASH TABLES: OBIETTIVI

- DHT: Obiettivi
  - Scalabilità
  - Flessibilità
  - Affidabilità

- Adattabilità a fallimenti, inserimento ed eliminazione di nodi
  - Assegnamento di informazioni ai nuovi nodi
  - Re-assegnamento e re-distribuzione delle informazioni in caso di fallimento o disconnessione volontaria dei nodi dalla rete

- Bilanciamento delle informazioni tra i nodi
  - Fondamentale per l'efficienza della ricerca

CONFRONTI TRA I DIVERSI APPROCCI

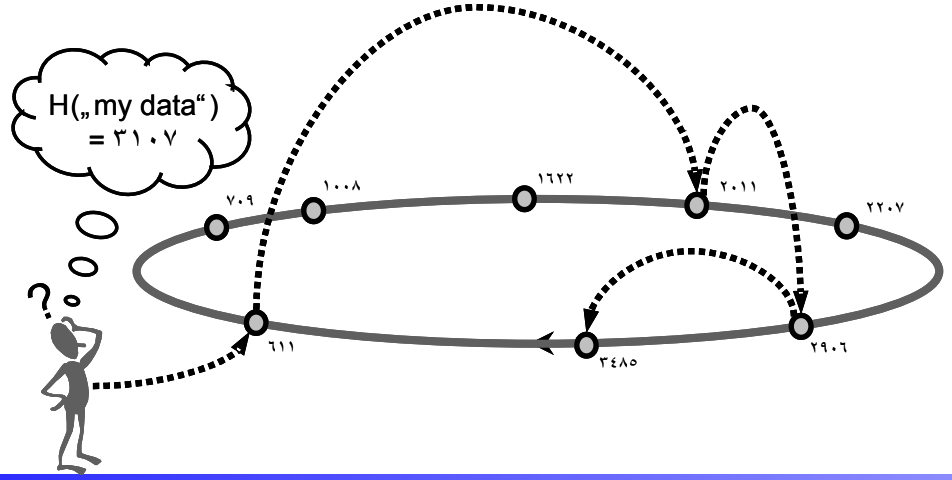
Approccio	Memoria per Nodo	Overhead di Comunicazione	Queries Complesse	Falsi Negativi	Robustezzza
Server Centrale	O(N)	O(1)	✓	✓	✗
P2P puro (flooding)	O(1)	$O(N^2)$	✓	✗	✓
DHT	O(log N)	O(log N)	✗	✓	✓

DHT: GESTIONE DISTRIBUITA DEI DATI

- Mapping dei nodi e dei dati nello stesso spazio di indirizzamento
  - Ai peers sono associati degli identificatori unici (ID), che li individuano univocamante all'interno del sistema
  - Anche ai dati sono associati degli identificatori unici che gli identificano univocamente nel sistema
  - Esiste uno spazio logico comune degli indirizzi per i dati e per i peer.
  - I nodi sono responsabili della gestione di una porzione dello spazio degli indirizzi corrispondente ad un sottoinsieme dei dati mappato in quello spazio
  - La corrispondenza tra i dati ed i nodi può variare per l'inserimento/cancellazione di nodi
- Memorizzazione/ Ricerca dei dati
  - Ricerca di un dato = routing verso il nodo responsabile
  - Ogni nodo mantiene una tabella di routing, che fornisce al nodo una visibilità parziale del sistema
  - Key based Routing:Il routing è guidato dalla conoscenza dell'ID del dato ricercato
  - Falsi negativi eliminati

DHT: INDIRIZZAMENTO

- Tecniche di indicizzazione distribuita
  - Spazio degli indirizzi in cui vengono mappati sia i dati che i nodi
  - I nodi intermedi mantengono delle tabelle di routing
    - Instradamento efficiente verso il nodo "destinazione"
    - Content routing (guidato dagli ID dei dati),
- Problemi
  - Gestione dinamica delle tabelle di Routing (inserzioni, eliminazioni)
  - Queries complesse non supportate (e.g, range query,ricerche contenenti wildcard)

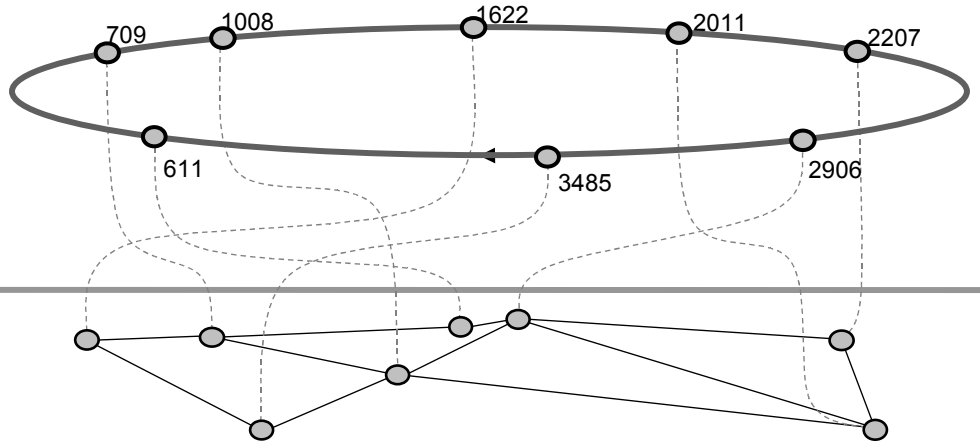


DHT: INDIRIZZAMENTO

Passo 1:

- Definizione dello spazio degli indirizzi logici. Esempio: Spazio degli indirizzi strutturato secondo un anello logico. S
  - Lo spazio lineare degli indirizzi logici 0, …, 2^m-1 è molto più grande del numero di oggetti da memorizzare (es m=160),
  - Sullo spazio degli è definito un ordinamento totale (operazioni in modulo)
- Associazione nodi-indirizzi indirizzi logici avviene mediante una funzione hash
- La topologia reale e logica (overlay network) non sono in genere correlate

Distributed Hash Table: Visione logica



Mapping sulla topologia reale

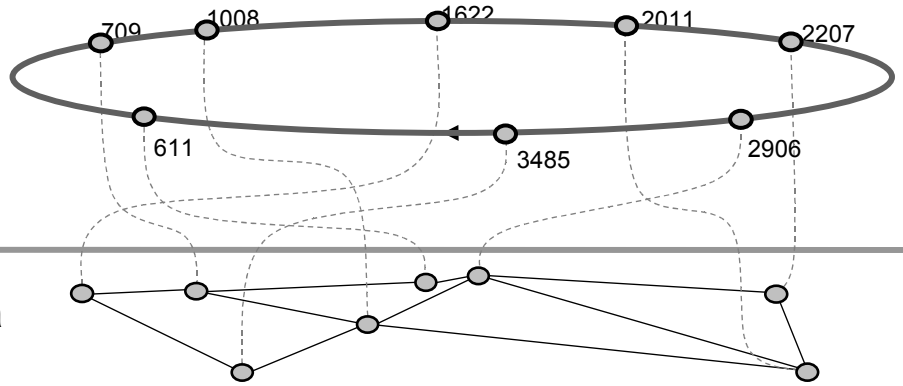
DHT: INDIRIZZAMENTO

Passo 2:

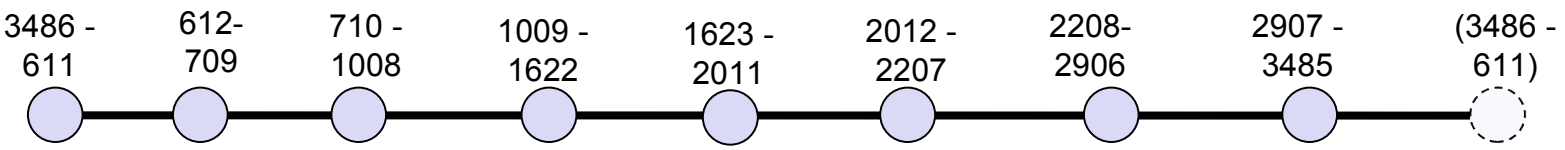
- Ogni nodo è responsabile di una parte dei dati memorizzati nella DHT
- In generale ad ogni nodo viene assegnata una porzione contigua dello spazio degli indirizzi.
- I dati vengono mappati nello stesso spazio degli indirizzi dei nodi, mediante la funzione hash
  - E.g., Hash(*String*): H(„ LucidiLezione29-02-08 “) → 2313
  - Esempi: hashing del nome del file o del suo intero contenuto
- Ogni nodo memorizza informazioni relative ai dati mappati sulla propria porzione di indirizzi
- Spesso si introduce una certa ridondanza (overlapping)

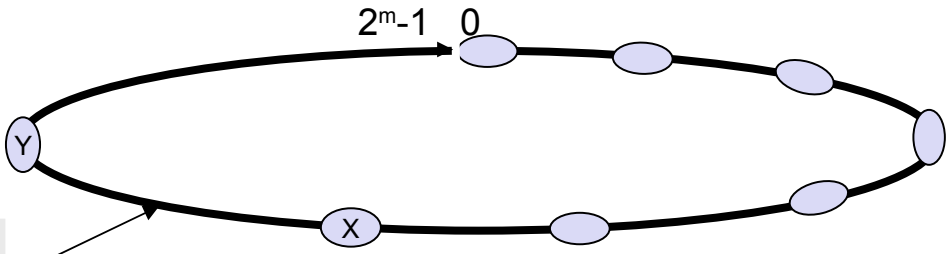
DHT: INDIRIZZAMENTO

Distributed Hash Table: Visione logica



Mapping sulla topologia reale





H(Node Y)=3485

Dato "D": H("D")=3107

H(Node X)=2906

DHT: BILANCIAMENTO DEL CARICO

- Problema: distribuzione uniforme degli indirizzi tra i peers che definiscono la DHT

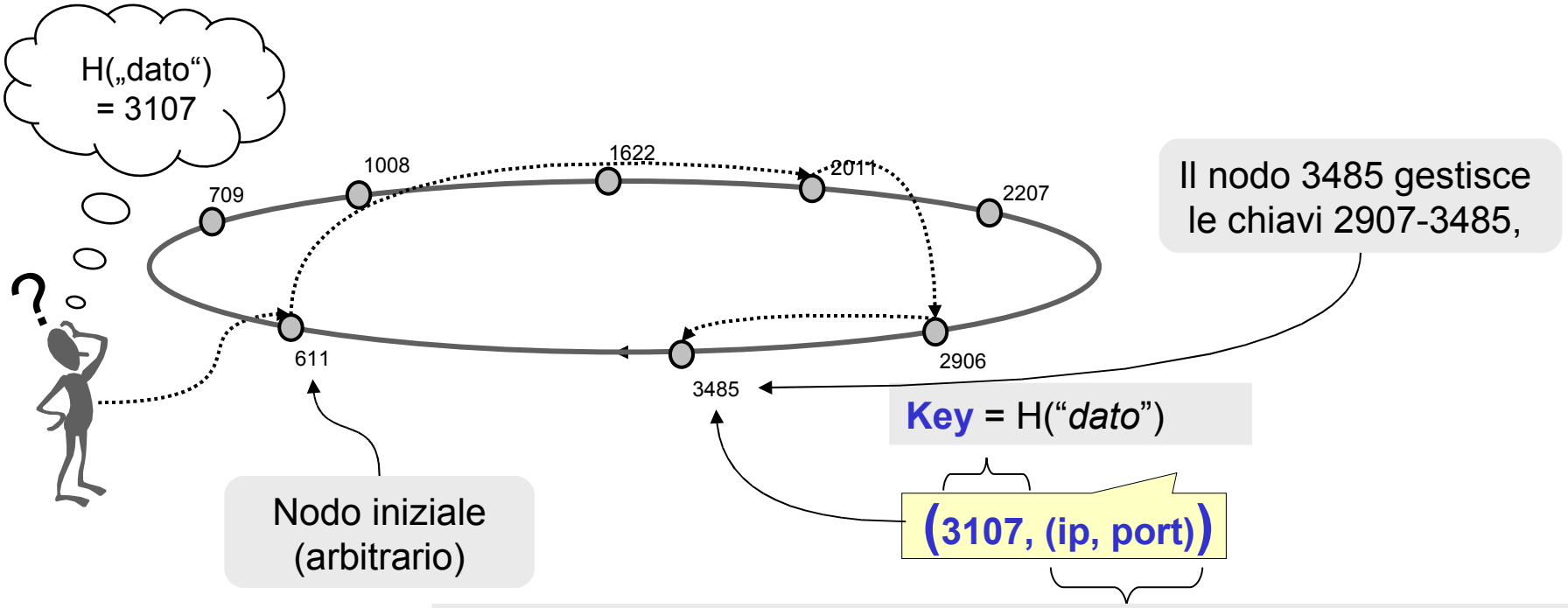
- Cause di possibili sbilanciamenti del carico:
  - Un nodo deve gestire una grossa porzione dello spazio degli indirizzi
  - Gli spazi degli indirizzi sono distribuiti in modo uniforme tra i nodi, ma gli indirizzi gestiti da un nodo corrispondono a molti dati
  - Un nodo deve gestire diverse queries, perché i dati corrispondenti agli indirizzi gestiti sono molto richiesti

- Sbilanciamento del carico comporta minor robustezza del sistema, minor scalabilità. Complessità O(log N) non garantita

- Definiti algoritmi di bilanciamento del carico

DHT: ROUTING

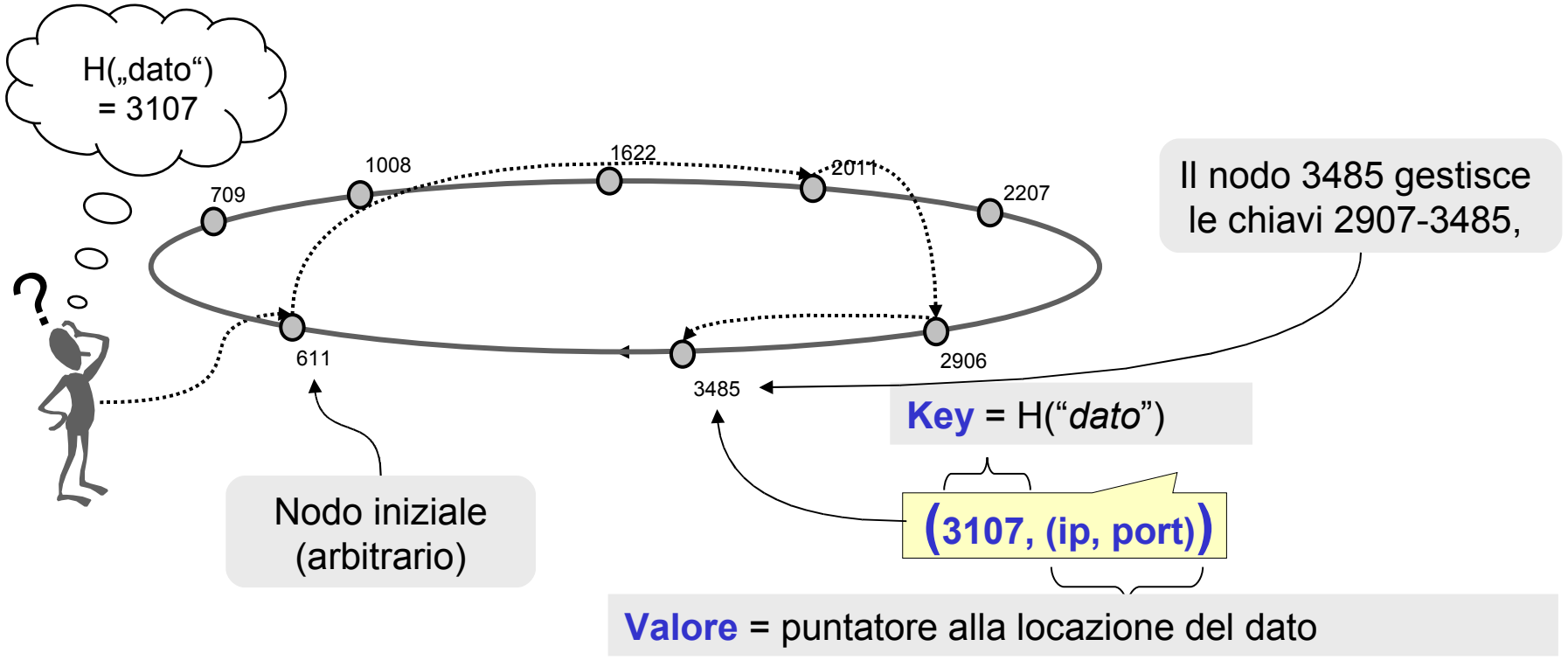
- Problema: dato D, cercare il nodo che gestisce key=H(D)
- La ricerca inizia in un nodo arbitrario della DHT
- La ricerca è guidata da H(D)



Valore = può essere un puntatore alla locazione effettiva del dato

DHT: ROUTING

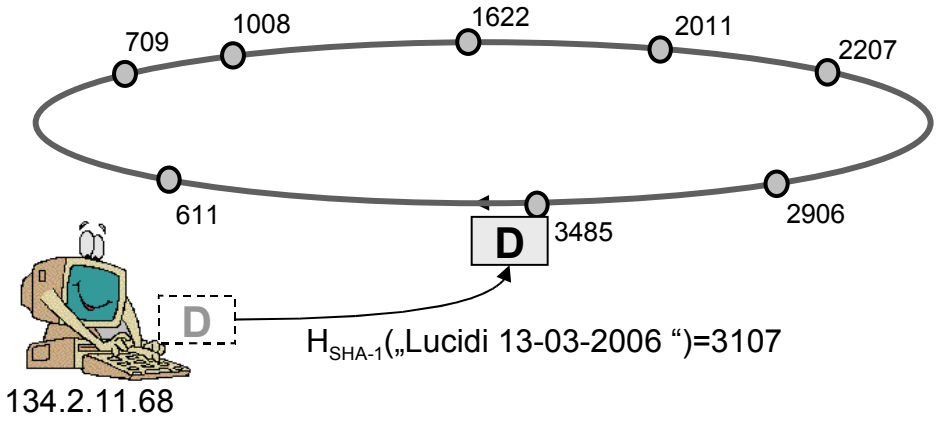
- Ogni nodo ha in genere una visione limitata della DHT
- Next hop: dipende dall'algoritmo di routing.
  - Esempio: basato sulla "vicinanza" tra l'ID del dato e l'ID del nodo (routing content based). Sistemi non strutturati: routing basato solo su connessioni tra nodi vicini



DHT: MEMORIZZAZIONE DIRETTA

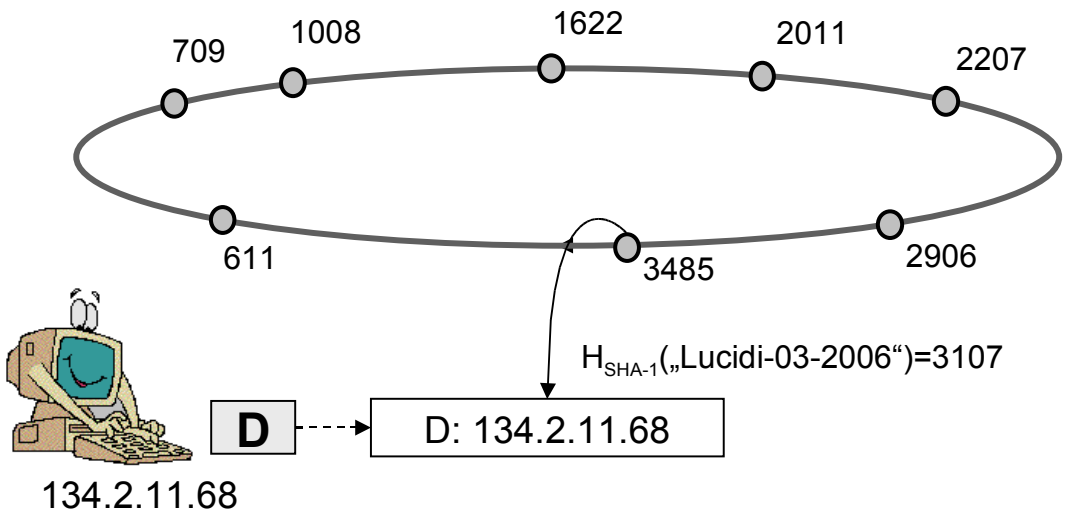
- La DHT memorizza coppie del tipo (key,valore)
- Valore = valore del dato ricercato
- Il dato viene copiato, al momento del suo insermento nella DHT, nel nodo che ne è responsabile (secondo il mapping dati-nodi). NOTA BENE: tale nodo non è in generale il nodo che ha inserito il dato nella DHT.

  Esempio: key = H("*Lucidi 13-03-2006* ") = 3107 . Il dato viene memorizzato sul nodo responsabile dell'indirizzo 3107.



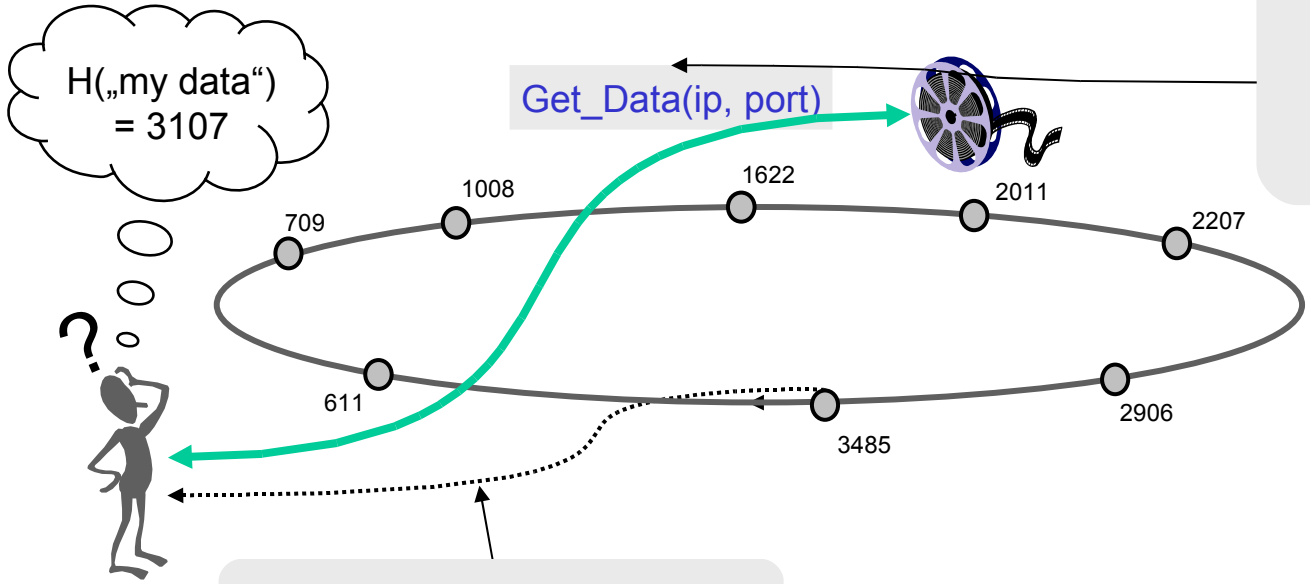
DHT:MEMORIZZAZIONE INDIRETTA

- Valore = riferimento al dato ricercato (es: indirizzo fisico del nodo che memorizza il contenuto)
- Il nodo che memorizza il dato può essere quello che lo ha inserito nel sistema
- Più flessibile, richiede un passo in più per l'accesso al dato



DHT: MEMORIZZAZIONE INDIRETTA

- **Trasferimento dei dati**
  - Si spediscono indirizzo IP e porta al richiedente
  - Il richidente effettua il download dei dati

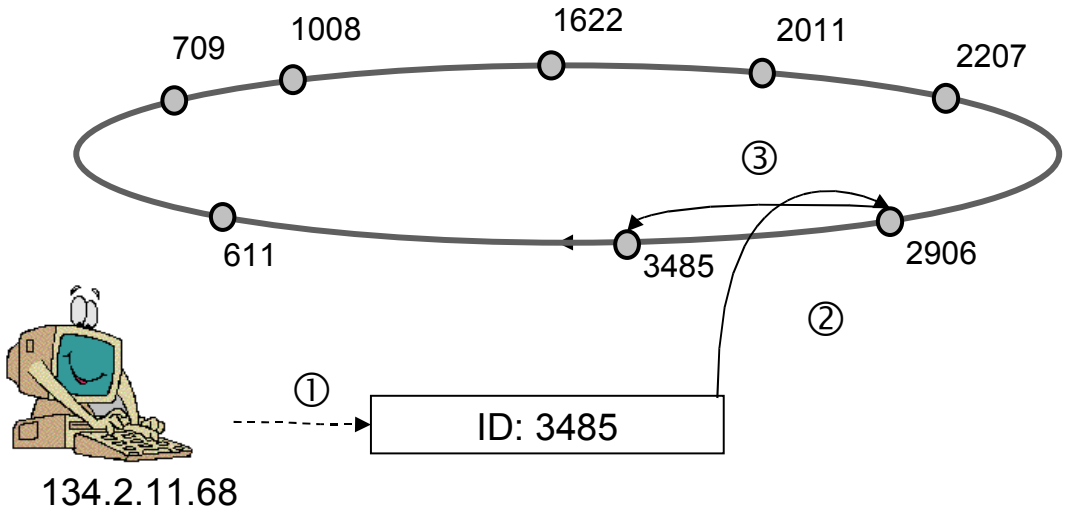


Memorizzazione Indiretta:
dopo aver ricevuto l'esatta
locazione del dato,
Il dato viene scaricato

Il nodo 3485 invia
(3107, (ip/port)) al richiedente

DHT:INSERZIONE DI NUOVI NODI

- Calcolo dell' identificatore del nodo, ID
- Il nuovo nodo contatta un nodo arbitrario della DHT (bootstrap)
- Assegnamento di una porzione dello spazio degli indirizzi ad ID
- Copia delle coppie K/V assegnate (in genere si utilizza ridondanza)
- Inserzione nella DHT (collegamento con nodi vicini)



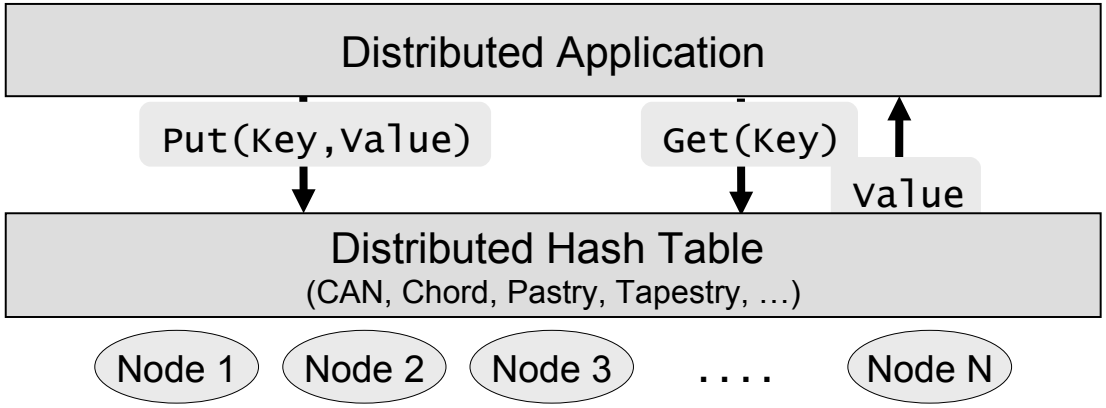
DHT:RITIRO/FALLIMENTO DI NODI

- Ritiro Volontario di un nodo
  - Partizionamento della propria porzione degli indirizzi sui nodi vicini
  - Copia delle coppi chiave/valore sui nodi corrispondenti
  - Eliminazione del nodo dalle tabelle di routing

- Fallimento di un Nodo
  - Se un nodo si disconnette in modo inatteso, tutti i dati memorizzati vengono persi a meno che non siano memorizzati su altri nodi
    - Memorizzazione di informazioni ridondanti (replicazione)
    - Perdita delle informazioni; refreshing periodico delle informazioni
  - Utilizzo di percorsi di routing alternativi/ridondanti
    - Probing periodico dei nodi vicini per verificarne la operatività. In caso di fault, aggiornamento delle routing tables

DHT: APPLICAZIONI

- Interfaccia (API) per l'accesso alla DHT
  - Inserimento di Informazione Condivisa
    - PUT(key,value)
  - Richiesta di Informazione (content serach)
    - GET(key)
  - Risposte
    - Value
- L'interfaccia è comune a molti sistemi basati su DHT



DHT: APPLICAZIONI

- Le DHT offrono un servizio generico distribuito per la memorizzazione e l'indicizzazione di informazioni

- Il valore memorizzato in corrispondenza di una chiave può essere
  - Un file
  - Un indirizzo IP
  - O qualsiasi altro dato……

- Esempi di applicazioni che possono utilizzare le DHT
  - Realizzazione di DNS
    - Chiave: hostname, valore: lista di indirizzi IP corrispondenti
  - P2P storage systems: es. Freenet

  ……

CONCLUSIONI

Proprietà delle DHT

- Il routing è basato sul contenuto della query
- Le chiavi sono equamente distribuite tra i nodi della DHT
  - Si evitano i colli di bottiglia
  - Supportano l'inserzione incrementale di chiavi nel sistema
  - Tolleranti ai guasti
- Sistemi auto-organizzanti
- Realizzazione semplice ed efficiente
- Supportano un ampio spettro di applicazioni
  - I valori associati alle chiavi dipendono dalla applicazione

DHT: SISTEMI ESISTENTI

- Chord

  UC Berkeley, MIT
- Pastry

  Microsoft Research, Rice University
- Tapestry

  UC Berkeley
- CAN

  UC Berkeley, ICSI
- P-Grid

  EPFL Lausanne

- … ed altri: Kademlia, Symphony, Viceroy, …