

Lezione n.11
APPLICAZIONI DI DHT:
LA RETE KAD DI E-MULE E
BITTORRENT

Laura Ricci
22/3/2013

LA RETE KAD

- La rete KAD: caratteristiche generali
- Tabella di routing e gestione dei bucket: confronto con Kademlia
- Pubblicazione dei dati: 2-level model
- Ricerca di informazioni
- Gestione peer a monte di NAT: sistema dei buddy
- L'interfaccia

LA RETE KAD: CARATTERISTICHE GENERALI

- Implementazioni di Kademia:
 - **KAD**: DHT implementate in **Emule** e basate su Kademia, supportata dal client eMule (a partire dalla versione 0.40)
 - recentemente anche **BitTorrent** implementa un protocollo basato su Kademia
- Introdotta per limitare gli svantaggi della rete basata su servers
 - eliminare colli di bottiglia (30 servers gestiscono il 90% della rete)
 - migliorare l'anonimato
- Caratteristiche generali del protocollo KAD
 - implementazione basata su Kademia,, alcune modifiche rispetto alla proposta originaria
 - controllo automatico dell'accessibilità delle porte (NAT +Firewalls)

LA RETE KAD: CARATTERISTICHE GENERALE

- Spazio degli identificatori: 128 bits, distanza basata su metrica \oplus
 $2^{128} \approx 3.40 * 10^{38}$, identificatori possibili
- Tabella di routing basata su una modifica dei k-buckets di Kademia
 - la struttura dell'albero binario corrispondente alla tabella di routing è modificata in modo da memorizzare un maggior numero di contatti
- **bootstrap** basato sulla conoscenza di peer attivi sulla rete KAD. E' possibile
 - specificare l'indirizzo di un peer conosciuto
 - oppure usare una lista di client memorizzata sul client nella cartella config di eMule (client conosciuti nelle sessioni precedenti)
- gestione di peers a monte di firewall o NAT
 - firewall check
 - buddy peers

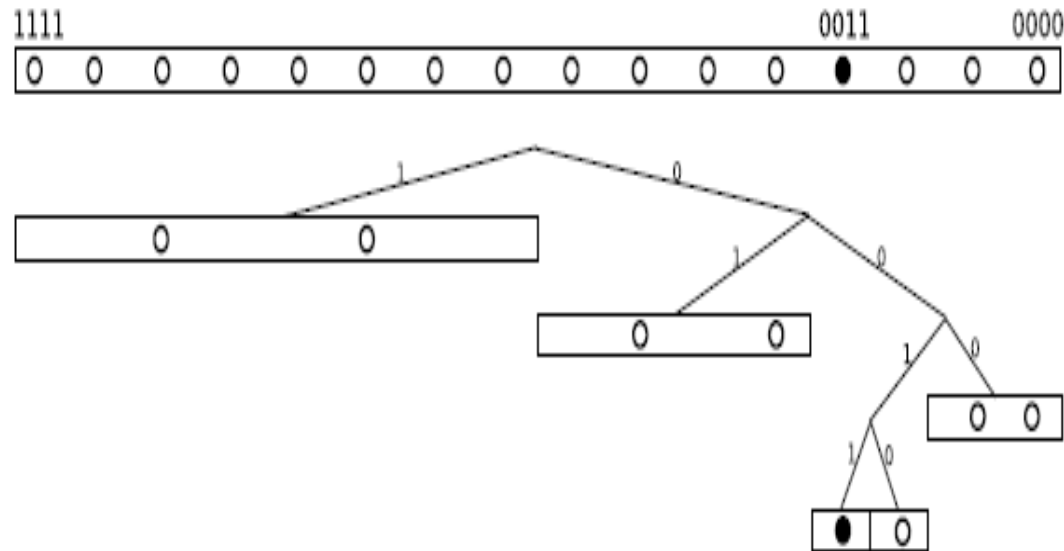
PUBBLICAZIONE DI FONTI

- La pubblicazione di un intero file o di parti di esso sulla DHT provocherebbe un alto tasso di traffico sull'overlay
- Per questa ragione, quando un peer intende **condividere** un file F
 - memorizza F su **un supporto locale**
 - **non utilizza** la DHT per pubblicare F o parti di F
 - utilizza la DHT per memorizzare un **insieme di riferimenti ad F**
- La pubblicazione dei riferimenti ad F consente un rapido reperimento di F, senza l'introduzione di un insieme di servers centralizzati
- In questo modo **ogni peer memorizza una parte dell' indice** contenente alcuni file condivisi in KAD
- replicazione dei riferimenti: ogni riferimento R ad F viene memorizzato su un insieme di nodi della DHT. Questo insieme di nodi definiscono la **tolerance zone** di R

CLIENT ID E FILE HASHING

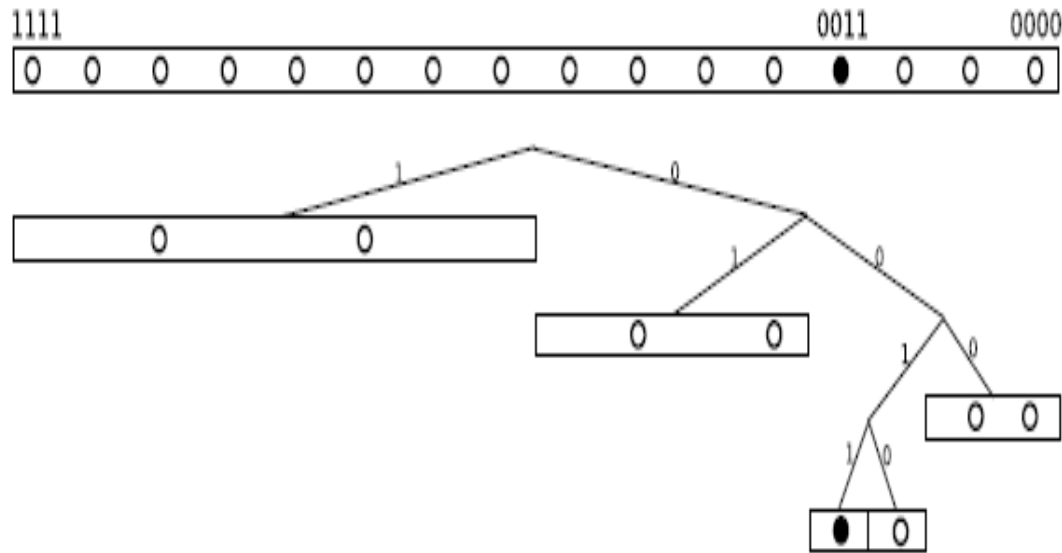
- Identificatori hash associati a client e a risorse
- KAD utilizza **identificatori di 128 bits**, invece dei 160 bit proposti originariamente in Kademlia
 - motivazione: compatibilità con eMule, che utilizza identificatori di 128 bit per gli identificatori dei file
 - si utilizza lo stesso algoritmo di hashing di eMule (basato su MD4)
- eMule possiede già un sistema per identificare i clients, cioè il client hash utilizzato per implementare il sistema dei crediti
 - contiene alcuni byte con valore prefissato, introdotti per distinguere il client eMule da altri client (eDonkey, MLDonkey)
 - non utilizzato da Kademlia perchè gli identificatori non sono distribuiti uniformemente
 - **KadID**: Identificatore utilizzato da Kademlia, generato mediante MD4

KADEMLIA: LA TABELLA DI ROUTING



- la figura rappresenta la struttura della tabella di routing del peer nero
- tabella di routing: **albero binario non bilanciato**, le foglie contengono insiemi di identificatori caratterizzati da un prefisso comune
- il cammino dalla radice ad una foglia f rappresenta il **prefisso comune** a tutti gli identificatori associati a quella foglia

KADEMLIA: LA TABELLA DI ROUTING



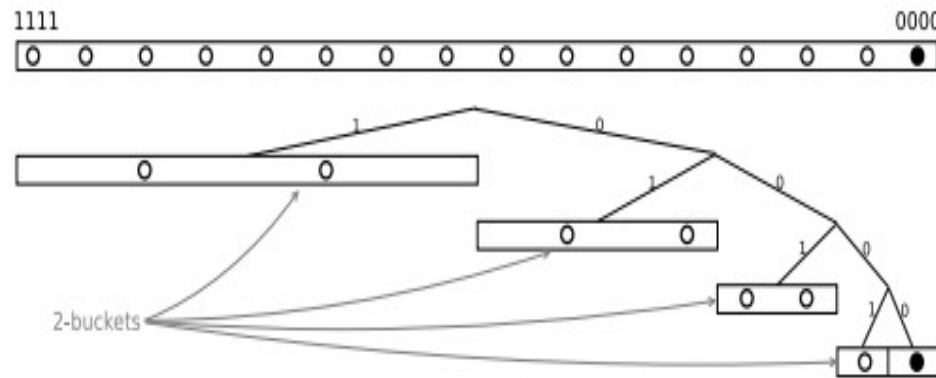
la figura rappresenta la struttura della tabella di routing del peer nero

le foglie corrispondono ai k-buckets della tabella di routing

Più alto il livello della foglia, minore il prefisso a comune tra gli identificatori associati alla foglia e quelli del peer considerato (il peer nero)

L'albero viene linearizzato e memorizzato in una tabella

KAD: LA TABELLA DI ROUTING



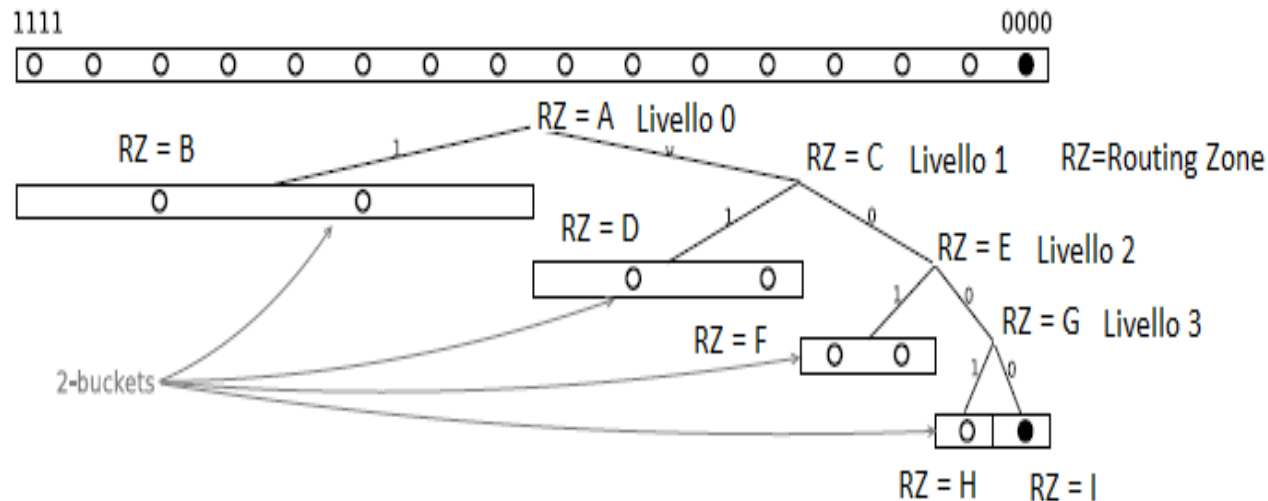
L'albero di Kademia viene ristrutturato per facilitare le operazioni

Nodi della tabella organizzati secondo la distanza \oplus rispetto al nodo considerato, invece che rispetto ai prefissi comuni

Distanza \oplus di un nodo da se stesso = 0 (nodo più a destra nero)

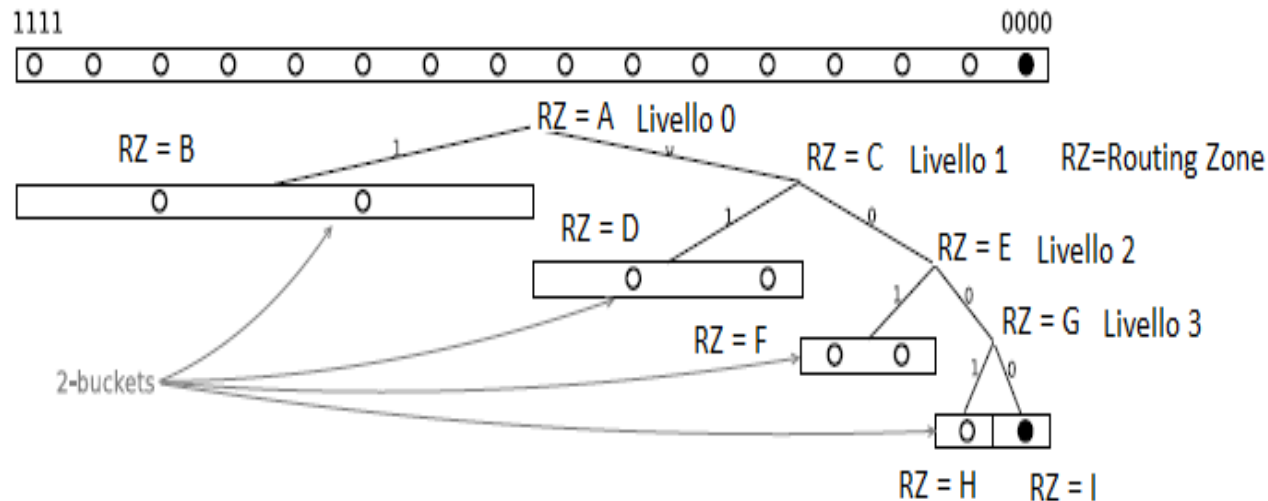
Cammino dalla radice ad una foglia individua una fascia di distanze \oplus tra nodi appartenenti al bucket associato alla foglia ed il peer considerato (peer nero)

KAD: LA TABELLA DI ROUTING



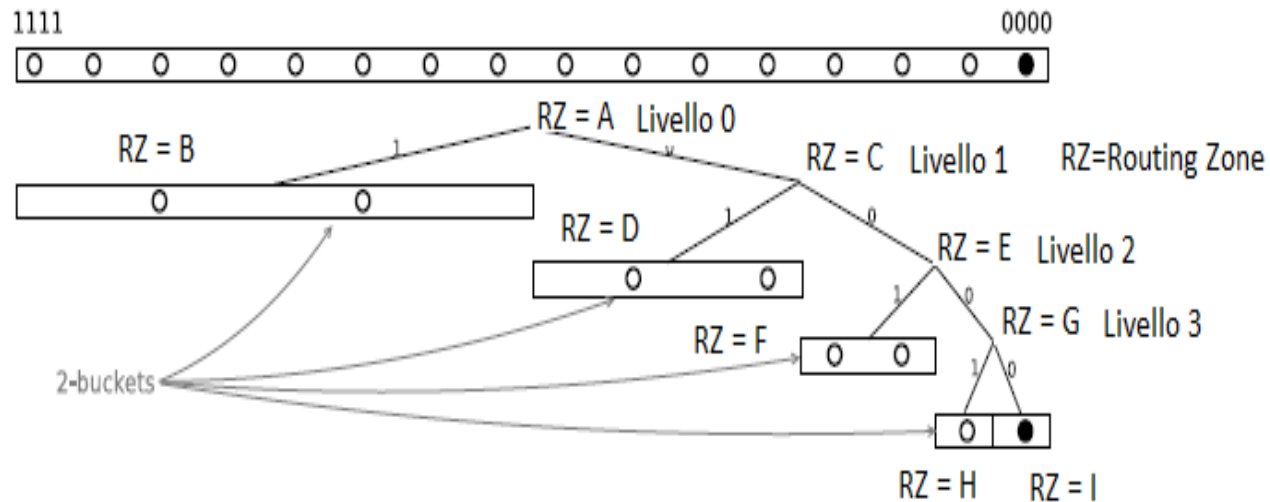
- **Albero delle distanze**: un cammino sull'albero identifica un **intervallo di distanze**. Facilita il processo di ricerca
- Le foglie corrispondono ai k-buckets della tabella di routings= routing bins, max 10 contatti per bin
- ogni k-bucket contiene nodi a distanza d rispetto al nodo considerato,
- L'albero **non rappresenta i prefissi dei nodi** contenuti nei buckets

KAD: LA TABELLA DI ROUTING



- RZ = Routing Zone
- Per ogni livello dell'albero due routing zones (potranno essere più di due)
- Associati ad ogni routing zone
 - livello
 - zone index: distanza in termine di numero di routing zones tra la zona considerata e la routing zone più a destra sullo stesso livello
 - utilizzato in seguito nel processo di espansione dell'alberi

ALBERO DELLE DISTANZE



Zone Index (A,C,E,G) = 0

Zone Index (B,D,F) = 1

AGGIUNTA DI NUOVI CONTATTI

- La strategia base di Kademia per l'aggiunta di nuovi contatti è stata modificata in modo da **aumentare il numero di contatti posseduti da un nodo**
- Aggiunta di un contatto
 - selezione del bucket (bin) appropriato
 - se il contatto C non è già stato inserito nel bin
 - c'è ancora spazio all'interno del bin, si inserisce C in coda
 - il bin è pieno ed è ammessa l'**espansione del bin**
 - duplicazione del bin ed inserimento del nuovo contatto. Nessun contatto viene scartato
- ma...l'espansione non controllata dell'albero degli identificatori porterebbe alla memorizzazione di un numero eccessivo di contatti
 - '**espansione controllata**' dell'albero dei contatti

AGGIUNTA DI NUOVI CONTATTI

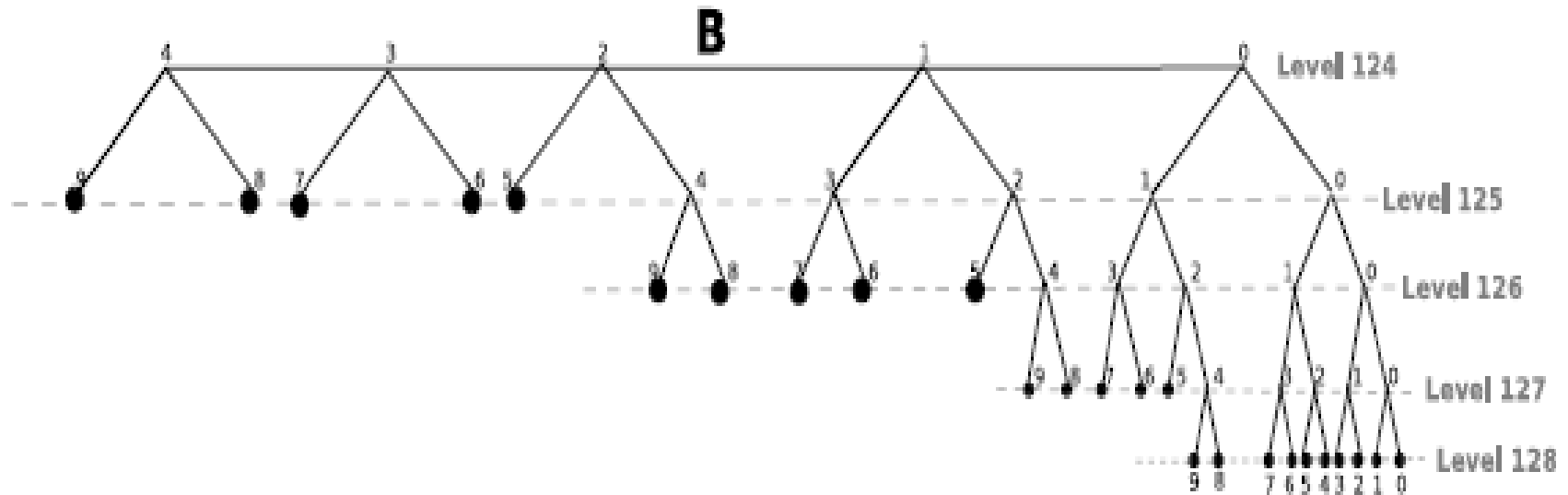
- per controllare l'espansione dell'albero si adotta la seguente politica:
 - memorizzare un maggior numero di contatti per routing zone 'vicine'
 - abilitare la suddivisione dei bins solo per tali zone
 - come in Chord, questa strategia consente un **routing logaritmico**
- duplicare un bin implica
 - trasformare una foglia in un nodo interno e associare i nuovi bin alle due foglie generate
 - aggiungere un livello all'albero
- Politica di espansione controllata dell'albero degli identificatori e quindi della routing table:
 - si ammette la duplicazione di un bin(foglia) solo per certi livelli dell'albero e per certe routing zones
 - per implementare questa politica, si utilizzano **Level** e **Zone Index** associate ad ogni Routing Zone

ESPANSIONE CONTROLLATA

La politica di espansione dell'albero degli identificatori è la seguente:

- Bin (foglia) di Livello ≤ 3 la suddivisione è sempre permessa.
 - si generano routing zone con zone index > 1
 - la suddivisione può portare ad ottenere un albero binario completo di profondità 4 con 16 bins a livello 4
- Bin (foglia) di Livello ≥ 4 : il bin si può suddividere solo se Zone Index < 5
 - nei livelli dell'albero di profondità maggiore di 4, solo i primi 5 bins possono essere partizionati
 - si suddividono i bins alla estrema destra dell'albero
 - questi bins contengono contatti relativi a zone più vicine al nodo preso in esame
- La suddivisione non può comunque superare il livello 127

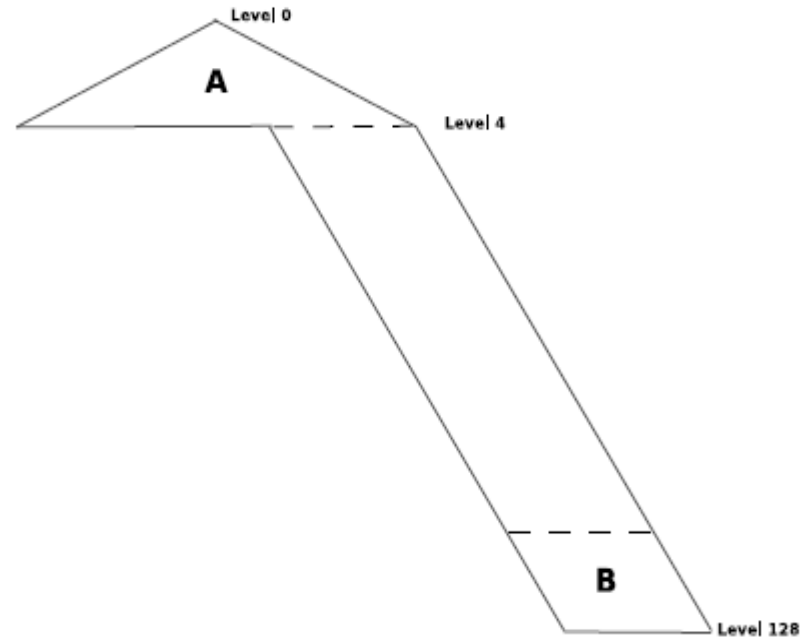
UN ALBERO KAD COMPLETO



Numero totale di contatti

$$\begin{aligned} \text{totale contatti} &= (\text{numero totale di buckets}) * K \\ &= (\text{foglie di livello 4} + \text{foglie dal livello 5 al livello 127} + \text{foglie a livello 128}) * K \\ &= (11 + 123 * 5 + 10) * 10 = 6360 \end{aligned}$$

UN'ALBERO KAD COMPLETO



Forma generale di un albero completo KAD

Le zone A e B sono mostrate in maggior dettaglio nei lucidi precedenti

Albero altamente sbilanciato a destra

Un maggior numero di contatti a distanze piccole

Caratteristica simile a Chord, conseguenza: routing logaritmico

GESTIONE DEI BINS

- La gestione delle routing tables di KAD considera il problema dell'alto livello di churn dei sistemi P2P
- Le tabelle di routing devono essere continuamente aggiornate per tener conto della continua modifica della rete
- Necessari due meccanismi
 - Ricerca di nuovi contatti
 - Eliminazione contatti obsoleti
- Ricerca nuovi contatti:
 - periodicamente (una volta ogni ora) si verifica se vi sono dei bin scarsamente popolati (<3 contatti) o bin che possono essere espansi
 - si genera un ID random nello spazio di identificatori coperto dal bin e si invia una query con chiave uguale all'ID generato
 - La ricerca produce come risultato alcuni nodi con identificatore vicino ad ID (se esistono). Questi vengono inseriti nel bin.

GESTIONE DEI BINS

Eliminazione di contatti non più connessi utilizza expire time e type del contatto

- **expire time**: frequenza di controllo presenza del contatto. Indica ogni quanto si deve effettuare il controllo su quel contatto
- **type**: 5 livelli, indicano il tempo di permanenza del contatto della rete
 - tipo 0 : contatti presenti da più di due ore
 - tipo 1 : tra 1 e 2 ore
 - tipo 2 : meno di un'ora
 - tipo 3 : nuovo contatto, non è stata effettuata ancora alcuna verifica su di esso
 - tipo 4 : tutti i contatti che non hanno risposto al ping per almeno una volta

GESTIONE DEI BINS

eMule v0.50a

Disconnetti
Kad
Server
Trasferime...
Cerca
File Condi...
Messaggi
IRC
Statistiche
Opzioni
Strumenti
Aiuto

Contatti (199)

ID	T...	Distanza
308CDCFC87B453EE34794D1017F2F31D	4(9)	00011011000101001000011000011110001101111101101111000111100001101111011111001110010101111101010...
38E96FCB1A7882A90B930D6CE2FE17DD	4(9)	0001000001110001001101010101010101000010111000010110100000011100100000001101011010111001010...
2380E457A6E8BFAF1E6AA3BF842F10FE	4(8)	00001000000110001011111010110101000010110000111001010111000111110111010100100001010000101...
22ABE79D68B08AD88B0B746F0A0A45D1	4(4)	000010010011001110111101011111101100011011111000111010101000001001000100101010001001010010101...
2953CB2FA20F88326899A41D23C825AB	3(9)	00000001011001011100100011100110100010010011000000001110001011010101011000001111100001011100111...
294A12A0F7E839D375D689B3A23E0FBD	3(9)	000000101101001001001000010000100100011110000111101010110110110110110110010010001110111101001001...
29416AFE10A94FDBCE181736229C94D4	3(9)	0000001011011001001100000001110010100000110001101101101101101100110000110110000110011000111001100...
2944ED9389BAD13D708668358C9012F9	3(9)	00000010110110010110111011100010011100110101010101010101101100110001100000001110110011111...
2947F242A508ED99EDFD2C3985D24367	3(9)	00000010110111110101000101000000010101011001110111100111100010010111001100011010010101000011...
2392B0BA5857C90C9B10DC63B7DBF9C8	3(9)	000010000000101011101010010110001110000011100001011010110010001011000100011101011101010011001...
22EB90CC5948223E0942786035A8CE5	3(9)	00001001011100111100101000101110011101011111011000101100100101100100011000010100100000101111100...
3B810C13719D06F74E7DD5376DB51D6C	3(9)	0001000000011001010101111000111000001111000110100100100100111110001101110001110110001111001101...
3336F3EC0B732CC14AD84467A6F451B5	3(9)	0001100010101110101010010000111010111000011100111000101001100010010000110001000101001110101...
31A946C235F806E933F540CFC60AD76A	3(9)	000110100011000100011100001000001000010110010111100100101000000111110000011010110010011000110101...
308F276DD64DD7AB1791497D983B5605	3(9)	0001101100010111011110110001111011001100010001001000011110000111101010000001111001011110000111...
3059A01837B32857EB13D1FA7A01C8A6	3(9)	000110111100000111110101111010100001111101110010111000011111001010001000111011011100000000...
30563241E94DC905B30598C258404860	3(9)	00011011110011100101000101000110101100100100010010110101101010110000100110111111000111000...
0A9C7D2B7C06E5978DE87B21F4A1456C	3(9)	001000010000010000100111110010011100110001101001011100011111110100111001110110000111011101011...

Ricerche Correnti (6)

Numero	Chiave	Tipo	Nome	Stato	Carico	Pacchetti Inviati	Risposte
290	6C8AD2085B873FB8FB93781430B82084	Ricerca Fonti	Peer to peer.pdf	In Arresto	0 (0 0)	14 7	0
291	1B0E64E1B44474B5DAEC63722ADC3C7D	Ricerca Fonti	Introduction to Queueing Theory - North Holland ...	In Arresto	0 (0 0)	23 8	3
-1	231B3F72DAD5F54CDA0AD20BBA63770D	Analisi Nodo		In Arresto	0 (0 0)	5 0	1
-1	3D3A0515A6697CC2FE1D1F49C6C0A768	Analisi Nodo		In Arresto	0 (0 0)	2 0	1

Ricontrolla Firewall
Disconnetti

BootStrap
 IP o Indirizzo: Porta:
 Carica 'nodes.dat' dall'URL:
 Dai Client Conosciuti
BootStrap

Contatti

Kad Rete

GESTIONE DEI BINS

- I contatti più 'vecchi' vengono favoriti, perchè maggiore è il tempo in cui un contatto è rimasto online, maggiore è la probabilità che esso rimanga online in futuro
- Algoritmo di **eliminazione** ed **aggiornamento dello stato dei contatti**:
 - più a lungo un contatto è rimasto sulla rete, meno di frequente viene contattato
 - i nuovi contatti inseriti, vengono immediatamente controllati
 - periodicamente (una volta al minuto) i contatti nei bin vengono analizzati e se l'expire time è scaduto il contatto viene controllato
 - si invia un messaggio di HELLO-REQ al contatto
 - se il contatto risponde, viene posto in coda alla lista del bucket, viene aggiornato il suo tipo e il suo expire time
- **Consolidamento dei bins**: controllo periodico dei bins contenenti meno di 5 contatti che vengono fusi (ogni 45 minuti)

GESTIONE DEI BINS

- Tabella aggiornamento stato di un contatto

Uptime	New type level	New expire time
0-1 h	Type 2	1 h
1-2 h	Type 1	1,5 h
> 2 h	Type 0	2 h

I contatti 'conosciuti' da più tempo vengono contattati meno di frequente

PUBBLICAZIONE DEI DATI: 2-LEVEL MODEL

- ad ogni file ed ad ogni client vengono assegnati **KadID**
- per ogni file vengono pubblicati due diversi tipi di coppie KadId-lista di puntatori

Informazioni di primo livello:

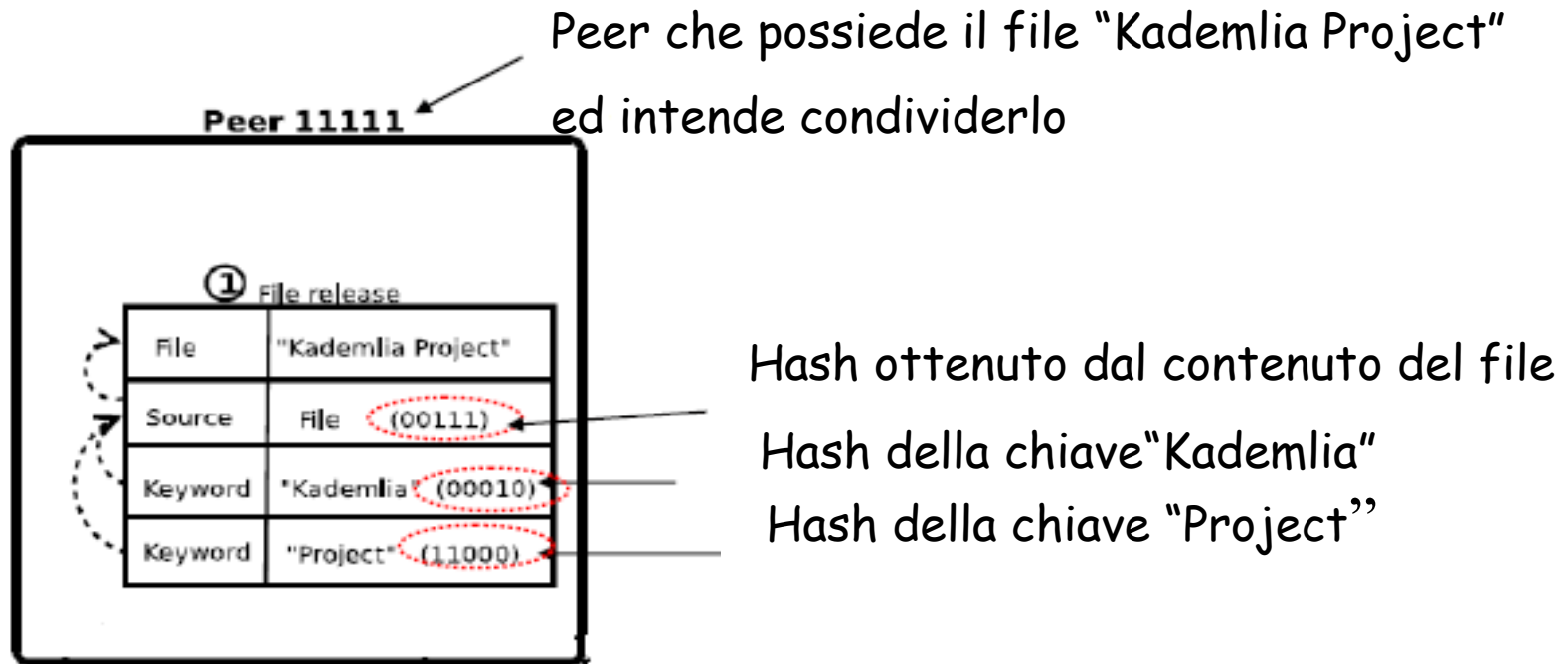
- **FileID**: hash del contenuto del file
punta a
- **Source**: identificatore logico del peer (nello spazio di identificatori della DHT) che ha pubblicato il file e che lo memorizza

Informazioni di secondo livello: Metadati

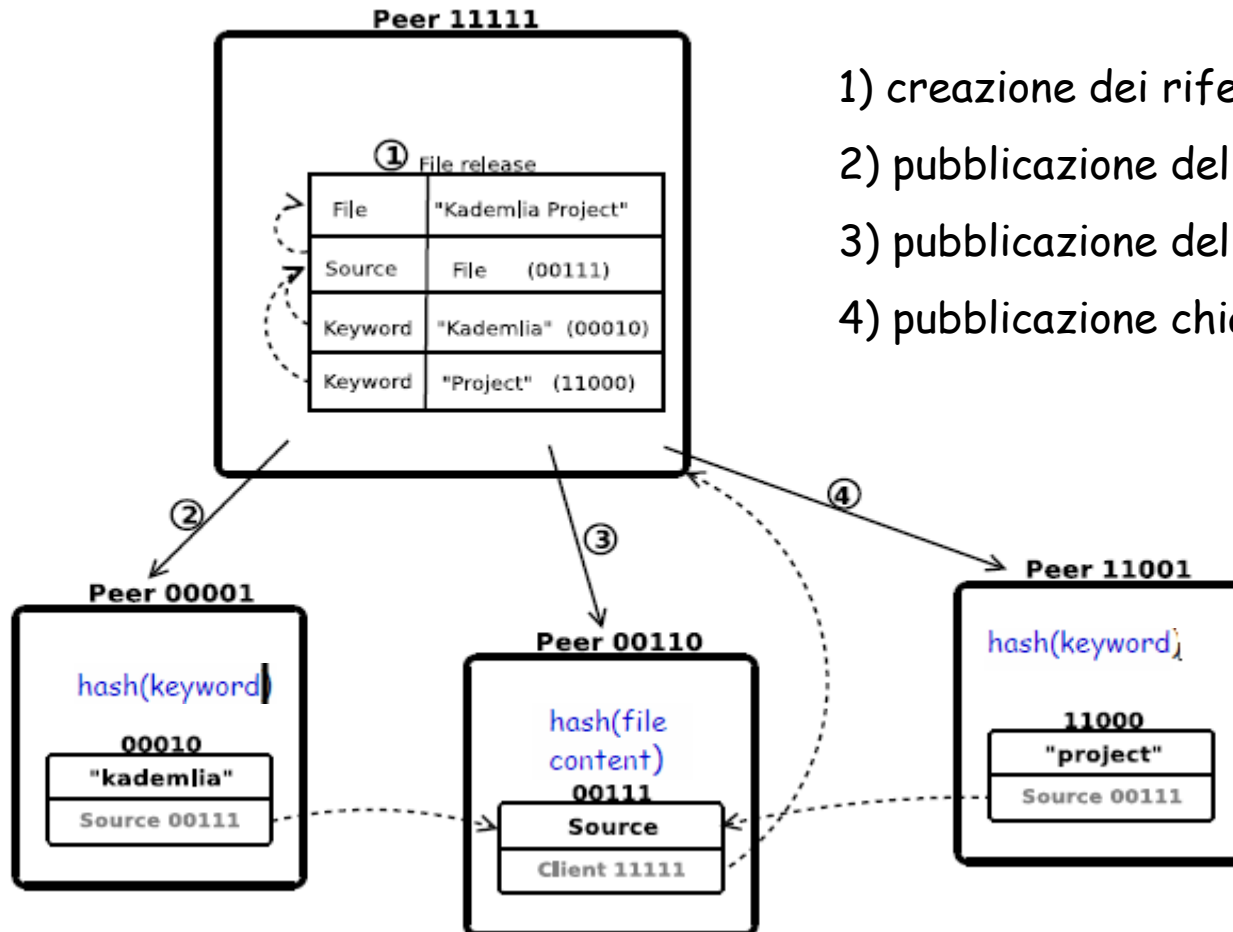
- **Keyword**: hash di una parole chiave che identifica il file
punta a.....
- **FileID**: hash del contenuto del file

2-LEVEL PUBLISHING MODEL

- Supponiamo di avere un file "Kademlia Project" che intendiamo condividere
 - 2 chiavi **kademlia project**

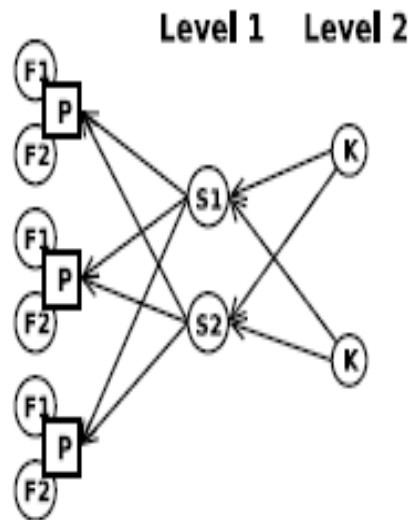


2-LEVEL PUBLISHING MODEL



- 1) creazione dei riferimenti
- 2) pubblicazione della chiave "kademlia"
- 3) pubblicazione della locazione del file
- 4) pubblicazione chiave "project2"

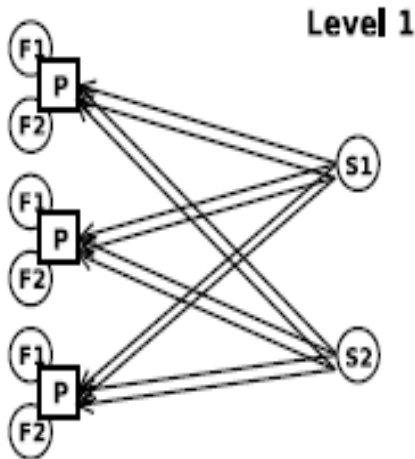
2-LEVEL PUBLISHING: VANTAGGI



- esistono due file diversi, F1,F2. F1 contiene la prima lezione di P2P, F2 contiene la seconda lezione di P2P
- ogni file è individuato dal 2 parole chiavi **dispensa, P2P**
- ognuno dei 3 peer pubblica entrambi i files, identificati dalle stesse parole chiave
- S1 contiene la Location Information per F1, S2 quella per F2
- **2 Level publishing:**
richiede 10 riferimenti

2-LEVEL PUBLISHING: VANTAGGI

1 Level Publishing



- vengono accorpati il livello dei metadati con quello della location information
- in un unico livello corrispondenza (parola chiave, Location Information)
- S1 contiene 3 riferimenti per la parola chiave 'dispensa' e 3 per la parola chiave P2P
- 12 riferimenti invece di 10

Se il tasso di replicazione dei files è alto il primo schema può ridurre notevolmente il numero dei riferimenti nella rete

2-LEVEL PUBLISHING MODEL

- Un riferimento di primo livello (Location Information) 'punta' direttamente ai client (sources) che hanno pubblicato e memorizzano il file
- Un riferimento di secondo livello (Metadati) punta ad un riferimento di primo livello
- I puntatore sono logici, cioè sono identificatori logici all'interno della DHT
- KAD pubblica anche, con lo stesso procedimento, le **note** che un utente associa ad un file
 - **rating** di un file
 - **commenti** sul file (es: nome che un utente ha dato al file)
 - la pubblicazione delle note utilizza lo stesso meccanismo
 - ad ogni nota viene associato un KadID e la nota viene mappata sulla rete KAD

2-LEVEL PUBLISHING MODEL

Nello schema presentato nel lucido precedente occorre considerare che

- ad una stessa **parola chiave** possono **corrispondere più files**
 - in generale è quindi necessario associare all'hash che identifica la parola chiave una lista di identificatori logici di files
- **lo stesso file** può essere messo in **condivisione da più utenti**
 - in generale è quindi necessario associare ad ogni file una lista di sorgenti . La lista contiene gli identificatori logici di tutti i peer che hanno messo in condivisione il file

KAD: PUBBLICAZIONE DI DATI

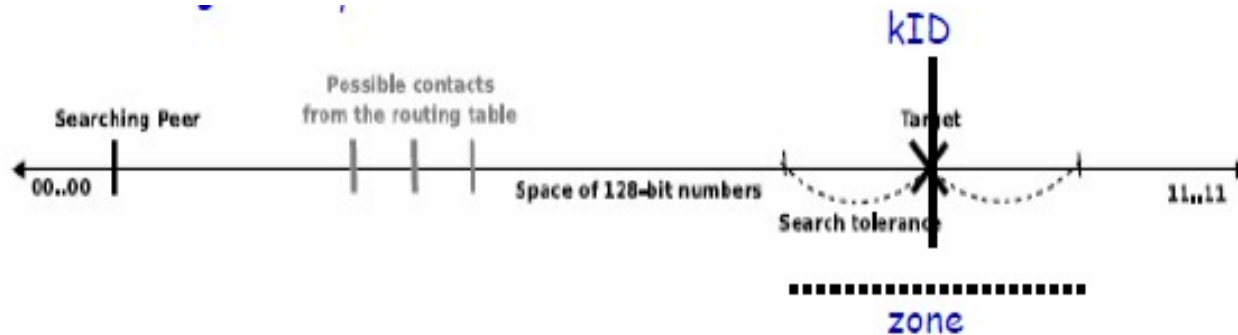
- Ogni nodo che pubblica un file deve
 - Pubblicare che esso stesso è una fonte per il file
 - Pubblicare le parole chiavi che identificano il file
 - Pubblicare eventuali note associate al file stesso
- **Pubblicazione di fonti:** il peer pubblica i riferimenti di **primo** e di **secondo livello** utilizzando come chiave, rispettivamente
 - l'identificatore del file
 - l'identificatore della chiave
- Una pubblicazione inizia sempre con il processo di ricerca del targetID che si intende pubblicare

CONTENT REPLICATION

Ogni informazione viene replicata su un insieme di nodi

- meccanismo utilizzato per far fronte all'alto livello di churn che caratterizza questo tipo di reti
- insieme di nodi su cui l'informazione ID viene replicata: tolerance zone dell' ID
- tolerance zone: nodi il cui identificatore differisce da ID meno di un valore soglia
 - solo gli ultimi bit della stringa binaria di 128 bit possono essere diversi
 - informazione replicata almeno su 10 nodi

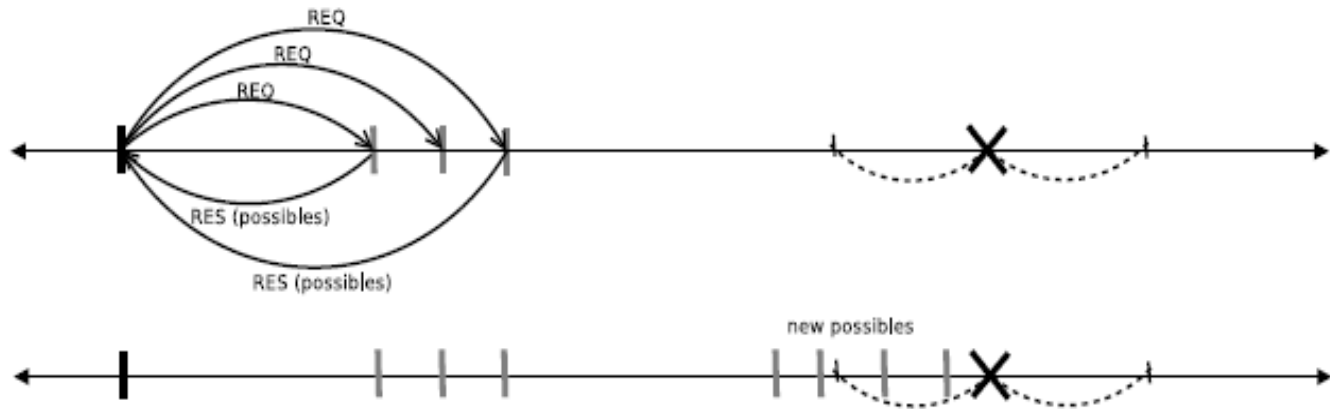
CONTENT REPLICATION: TOLERANCE ZONE



- Dove memorizzare una chiave KID?
 - nella sua **tolerance zone**: su 11 nodi i cui primi k ($k = 8$) bits sono uguali a quelli della chiave
 - tutti i peer in questa zona sono potenziali candidati per la memorizzazione della chiave
- La tolerance zone è centrata sul peer target **KID**

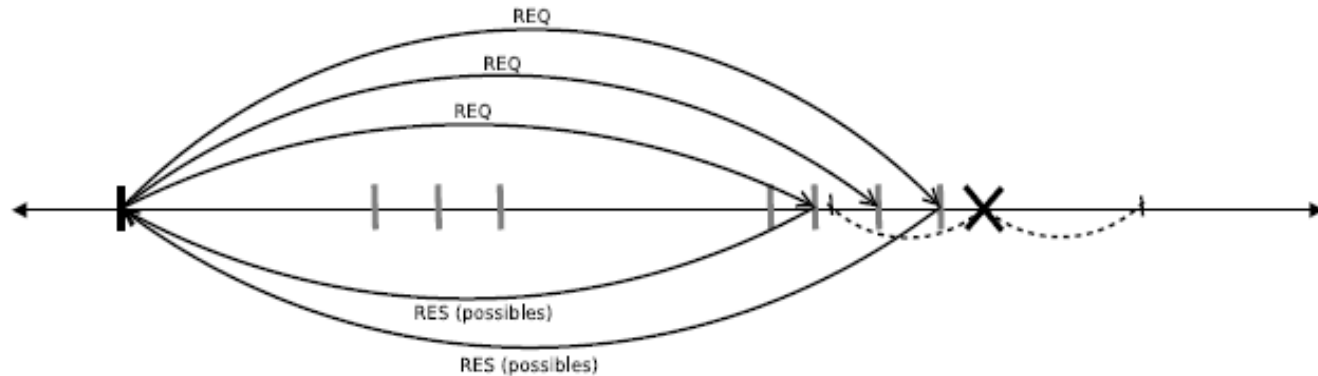
Il peer che intende memorizzare una nuova chiave ricerca nella propria routing table gli identificatori della tolerance zone

CONTENT REPLICATION: TOLERANCE ZONE



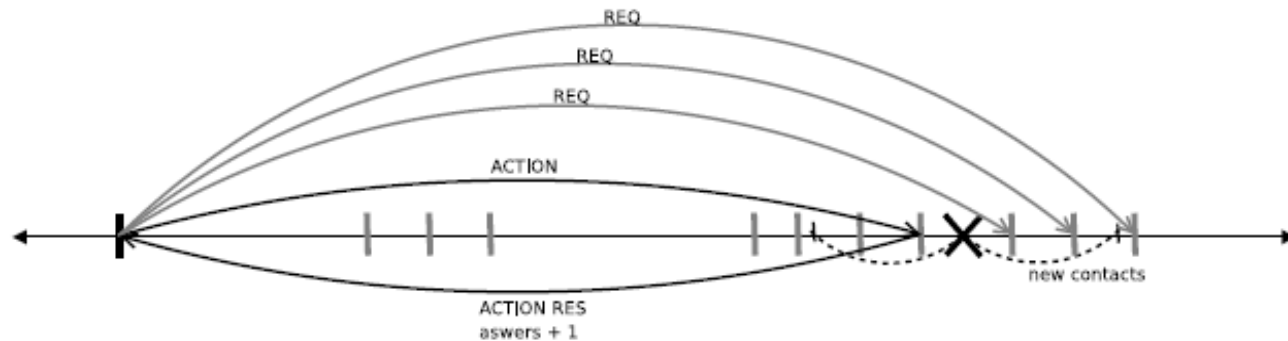
- Tutti i contatti in possesso del peer non sono ancora 'abbastanza vicini' alla chiave, cioè non appartengono alla zona di tolleranza
- i contatti che sono ancora presenti sulla rete, inviano al richiedente una lista di contatti **più vicini alla chiave**
- 4 nuovi contatti vengono comunicati al peer che effettua la ricerca durante la prima iterazione, due di questi si trovano nella tolerance zone della chiave

CONTENT REPLICATION: TOLERANCE ZONE



- tra i nuovi contatti di cui è venuto a conoscenza, il peer sceglie quelli più vicini al target e li contatta
- solo quelli presenti sulla rete rispondono comunicando nuovi contatti
- KAD memorizza su 11 nodi nella tolerance node, ma non necessariamente sul nodo più vicino alla chiave

CONTENT REPLICATION: TOLERANCE ZONE



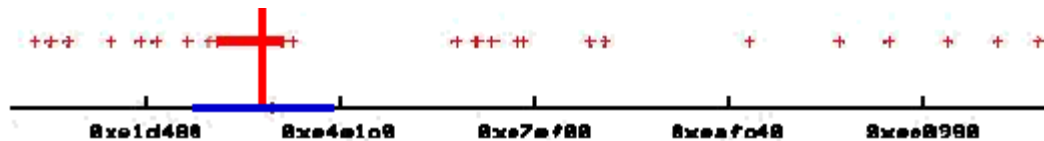
- quando il peer individua contatti attivi ed appartenente alla tolerance zone, invia al contatto il messaggio di action, ovvero una **richiesta di pubblicazione**

il contatto che ha eseguito l'azione invia un messaggio di ack. il peer incrementa un contatore e se il contatore ha raggiunto il massimo interrompe la ricerca

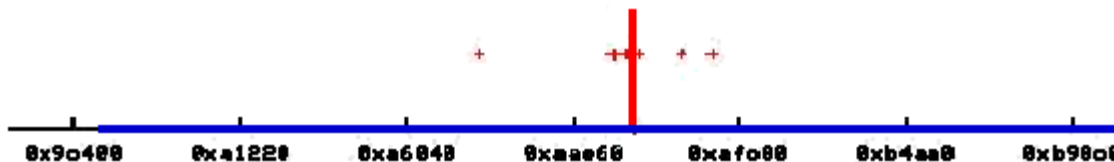
CONTENT REPLICATION: TOLERANCE ZONE

- Osservazione: una zona identificata da prefissi di 8-bits contiene circa 8000 nodi
- scenari di pubblicazione diversi a seconda che si utilizzi
 - una parola chiave molto popolare (molti peer la pubblicano)
 - una parola chiave rara (un singolo peer o pochi peer che pubblicano)

<the> pubblicata su 594 peer (97% all'interno della zona ad 8 bit)



<dreirad> pubblicata su 32 peer

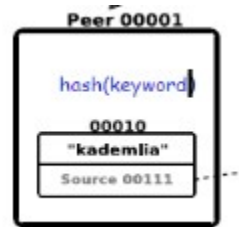


Parola chiave e' memorizzata in uno spazio di chiavi 30 volte più ampio

OVERLOAD PROTECTION

- Gli identificatori dei files sono distribuiti uniformemente sulla rete mediante la funzione hash
 - ogni nodo deve gestire approssimativamente lo stesso numero di entrate
- Esistono files con più popolari
 - alcuni nodi devono gestire più fonti ed altri con meno
 - diversità di interessi dei peer della rete comporta distribuzione abbastanza uniforme anche delle fonti
 - non esistono files condivisi da tutti...
- Invece, le parole chiave più popolari (ad esempio mp3) possono provocare hot spots nella rete
 - una parola chiave può identificare molti files
 - il peer che deve gestire tale parola chiave deve memorizzare un alto numero di riferimenti a file e gestire molte query

OVERLOAD PROTECTION: CHIAVI



- Ad ogni parola chiave possono corrispondere più informazioni di primo livello anche per lo stesso file, a causa della replicazione delle informazioni di primo livello su più peer.

Meccanismo di **Overload Protection**: un peer

non accetta ulteriori metadati se ne memorizza già una grande quantità

- ogni peer KAD può memorizzare al massimo **60000 keywords**
- Se un peer diventa un **hot spot** per una parola chiave molto popolare, rischia di utilizzare tutto lo spazio di memorizzazione a disposizione (i 60000 riferimenti) per quell'unica parola chiave
- In questo caso il peer è costretto a rifiutare la memorizzazione di riferimenti a parole chiave rare o poco conosciute

OVERLOAD PROTECTION: CHIAVI

Algorithm 4: Adding incoming keywords

Input: Metadata identified by a keyword

```
if total indexed keywords  $\leq$  60000 then
  if Keyword has already sources then
    if Keyword has already more than 50000 sources then
      | return a load of 100;
    else
      if SourceID does not exist OR total sources for this keyword  $\leq$ 
      45000 then
        | add source to Keyword;
        | total of indexed keywords + 1;
        | return load = (sources per Keyword*100)/50000;
      else
        | return a load of 100;
      endif
    endif
  else
    | add source to Keyword;
    | total of indexed keywords + 1;
    | return a load of 1;
  endif
else
  | return a load of 100;
endif
```

Un nodo non gestisce mai piu' di 60000 metadati

Un nodo non gestisce mai piu' di 50000 riferimenti per parole chiave

Se la chiave ha gia' piu' di 45000 Sorgenti, evitare di pubblicare le sorgenti che sono gia' molto popolari

Il valore load restituito al peer che ha pubblicato la parola chiave, influenza la frequenza con cui essa viene ripubblicata

OVERLOAD PROTECTION

- Meccanismo di protezione: supponiamo che il peer P pubblichi un file classificato secondo una parola chiave popolare
- Ogni peer a cui viene inviata la richiesta di pubblicazione risponde con un valore **load L**
 - numero di entrate già memorizzate per quella parola chiave
- P calcola la media di tutti i valori di L restituiti dai peer a cui ha inviato la richiesta di pubblicazione
 - media= valore medio di entrate memorizzate per quella parola chiave
 - se la media è superiore ad un valore soglia, la parola chiave è considerata popolare ed i nodi 'hot'
 - per 'proteggere' gli hot nodes, la parola chiave viene inserita in una 'banned list' e non ripubblicata più per un certo intervallo di tempo

OVERLOAD PROTECTION

- intervallo che deve trascorrere prima di una ripubblicazione della stessa chiave in funzione del valore di L
 - se $L < 20$ l'intervallo è di 24 ore
 - altrimenti intervallo = 7 giorni * (average load)/ 100
 - Esempio:
 - $L = 30$ (numero medio di riferimenti=15000), intervallo= 50 ore

OVERLOAD PROTECTION: SORGENTI

Algorithm 5: Adding incoming location information

```
Input: Incoming sources publish request identified by the sourceID
if Peer has already some sources then
  if a source with the same sourceID exists already from same peer then
    overwrite existing source;
    return load = (sources per sourceID*100)/300;
  else
    if Source has already 300 location information then
      remove oldest location information;
      add new source;
      return a load of 100;
    else
      add source;
      total of indexed sources + 1;
      return load = (sources per sourceID*100)/300;
    endif
  endif
endif
else
  add sources;
  total of indexed sources +1;
  return a load of 1;
endif
```



non piu' di 300 riferimenti
per ogni file

quando si raggiunge la soglia
si rimpiazza il riferimento più
vecchio

KAD: TIPI DI RICERCA



Node LookUp: si ricercano nuovi contatti per popolare la routing table.

Avviata periodicamente da e-Mule in modo automatico



Ricerca di Fonti: se esistono dei files nella coda di download, KAD ricerca automaticamente le fonti, cioè i peer che possono possedere quel file

Avviata periodicamente da e-Mule in modo automatico



Ricerca per parola chiave: avviata dall'utente che immette un insieme di parole chiave, KAD ricerca automaticamente i file che sono descritti da quelle parole chiave



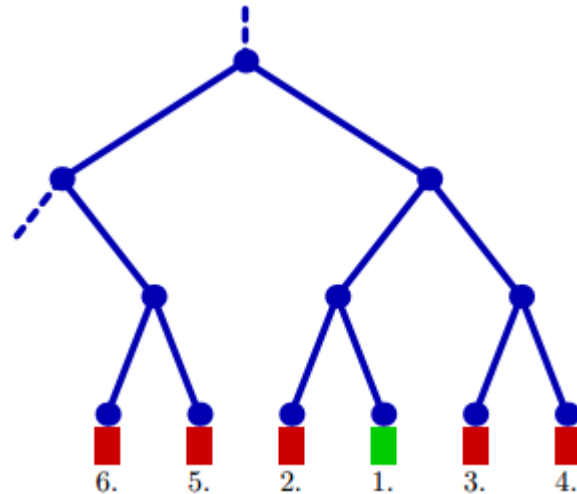
Ricerca di Note associate al file, avviata dall'utente

KAD: RICERCA DI INFORMAZIONI

- La ricerca riguarda in ogni caso un **target ID**
 - ID di un contatto generato casualmente
 - hash che identifica il file di cui si vogliono trovare le fonti
 - hash della parola chiave...
- Si ricercano nodi con ID 'vicino' al target ID
- Anche se l'espansione della tabella di routing consente di mantenere molti contatti, la probabilità di avere quello ricercato è comunque bassa, vista la dimensione dello spazio degli identificatori ed il numero di peer sulla rete
- Per ogni ricerca
 - un thread per il reperimento di nodi 'vicini' al target ID
 - un thread che interroga il nodo individuato dal primo thread per reperire/memorizzare informazione

KAD: RICERCA DI INFORMAZIONI

- Il primo thread ricerca nella tabella di routing i contatti più vicini (secondo la metrica XOR) al **target ID**
- Ricercati i 50 contatti più vicini
- Prima i 10 contatti nel bin più vicino al target ID, quindi i bins più vicini a questo e così via
- Ricerca nell'albero dei bins:



Bin verde rappresenta il più vicino al Target ID

RICERCA DI FONTI

Ricerca delle fonti: l'utente specifica almeno una parola chiave K

Primo passo: ricerca dei metadati

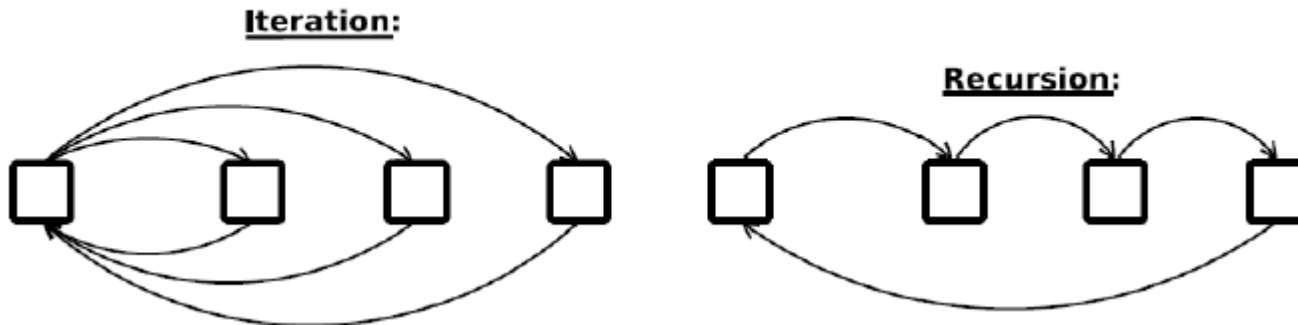
- si calcola l'hash H di K
- la chiave K viene ricercata nella **tolerance zone di H** , cioè nei peer con identificatore vicino ad H
- il client che ha iniziato la ricerca riceve un elenco di riferimenti a files che contengono quella parola chiave
- strategia di routing basata su Kademlia
 - iterativo
 - parallelo

Secondo passo: ricerca delle fonti (peer che posseggono il file)

- analoga alla ricerca dei metadati (routing iterativo + parallelo con zona di tolleranza)

KAD: RICERCA DI INFORMAZIONI

- Ricerca parallela con $\alpha = 3$
- Ricerca iterativa



KAD: RICERCA DI INFORMAZIONI

- Informazione inviata per nel messaggio di ricerca: targetID + numero di nodi richiesti
- Numero k di nodi richiesti dipende dalla informazione ricercata
 - Esempio: 11 nodi richiesti per la ricerca di nuovi nodi, solo 2 per la ricerca di fonti o di parole chiave
 - il nodo che riceve la richiesta di ricerca restituisce i k nodi più vicini al nodo richiesto
- La ricerca termina quando non viene restituito alcun nodo più vicino al target ID rispetto al nodo iniziale
- Gli identificatori via via reperiti sono utilizzati sia per continuare la ricerca che per popolare le routing tables

KAD: RICERCA DI FONTI

- Ogni client e-Mule possiede una coda di download per ogni file ricercato dall'utente
- Ricerca di nuove fonti per un file eseguita in background, per ogni file che si trova nella coda di download
- La ricerca di nuove fonti termina quando si hanno a disposizione per quel file 50 fonti diverse

DOWNLAOD DI FILES

Analogo a Emule e2k

- Basato sulla definizione di code di attesa nei clients
- Gestione dei crediti
- Protocollo TCP

LA RETE KAD: NAT/FIREWALLS

- Nella rete ed2K, il client Nattato o a monte di un firewall viene riconosciuto dal server che gli assegna un identificatore basso
- Esiste un meccanismo analogo per la rete KAD

Individuazione peer nattati: un peer P

- individua un contatto C sulla rete KAD
- viene effettuato **automaticamente il seguente check**
 - P invia a C un messaggio che include la porta TCP su cui P attende le connessioni
 - C individua l'indirizzo IP contenuto nel pacchetto ricevuto da P e lo rispedisce a P
 - P verifica se il suo indirizzo IP è uguale a quello ricevuto da C.
 - se i due indirizzi sono diversi P si trova a monte di un NAT
 - stato firewalled equivale al low ID della rete basate su servers

TRAFFICO KAD: ANALISI

Ogni riferimento è replicato k volte
... e ripubblicato periodicamente

Keywords: ogni 24 ore

Sorgenti: ogni 5 ore

Misurato il traffico di controllo in una zona "ad 8-bit" in un intervallo di 12 ore

Search

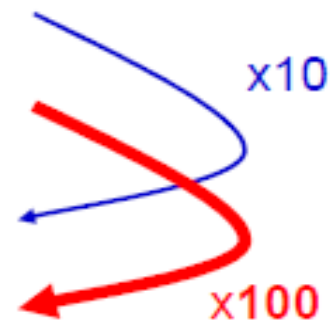
- ◆ Messages 561 542
- ◆ Traffic 10,8 Mbytes

Publish

- ◆ Messages 5 549 183
- ◆ Traffic 966 Mbytes

Route

- ◆ Messages 9 761 278
- ◆ Traffic 342 Mbytes



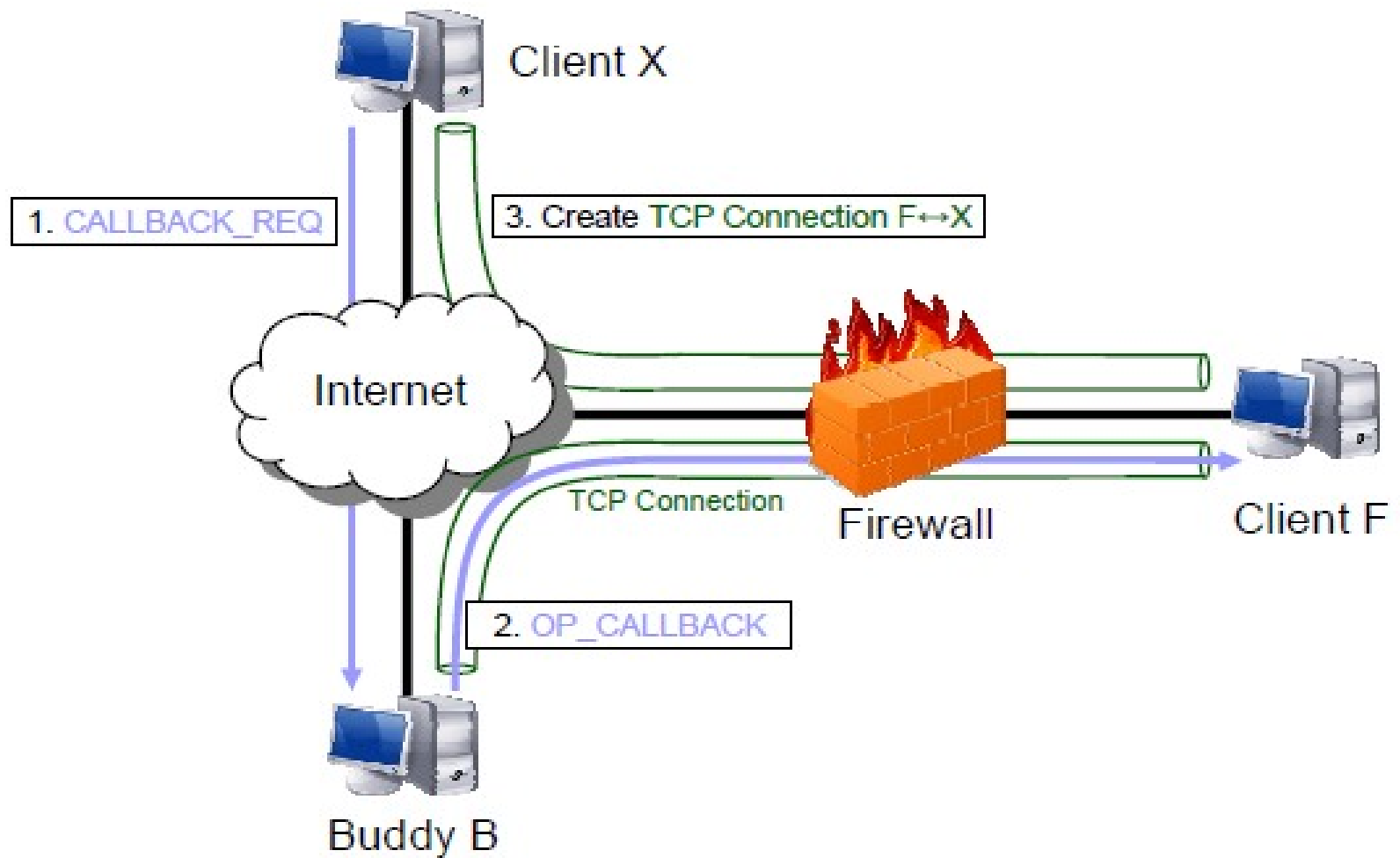
IL SISTEMA DEI BUDDY

- Un peer con stato firewalled non può ricevere richieste di pubblicazione/ricerca
- Se lo stato del peer risulta firewalled, KAD ricerca automaticamente un altro client con KAD non firewalled da utilizzare come compagno (**buddy**)
- Buddy peers = offrono supporto a peer in stato firewalled
 - il buddy riceve richieste per il peer firewalled e gliele reinoltra
- Ogni client con KAD non firewalled può gestire un numero massimo di un peer con stato firewalled

IL SISTEMA DEI BUDDY

- la ricerca dei buddy utilizza l'algoritmo per la ricerca dei contatti
- il target di tale ricerca è definito **invertendo i bit dell'identificatore** del peer che ricerca il buddy. Questo assicura
 - una distribuzione uniforme dei buddy
 - ogni nodo riceve al massimo una richiesta di diventare un buddy
- il buddy invia un messaggio di risposta al peer firewalled e questo apre una connessione verso il buddy (caso particolare di connection reversing)
- la connessione tra il buddy peer ed il firewalled peer è permanente
- un peer firewalled pubblica files e può essere comunque contatto da un altro peer che richiede il download di un file
- i peer firewalled non partecipano comunque alla rete di ricerca/pubblicazione

IL SISTEMA DEI BUDDY



KAD: L'INTERFACCIA

Contacts (670)

ID	Type	Distance
F940F53D956688A584E6487B7C9E7E7	0(0)	0111100111111011011101000111011010000110101100010101001001000...
64E988532B18F864E57A142CA6EA8EC	0(0)	111010100101001001101111101011110011011101100010101000101000...
38CEBF623CCDD099776005E86D61B	0(0)	10111011011010100110001100000111110110011011001101111100101010...
0F9C8C1454AA12AFC1F68219F118830C	0(0)	1000111100100110011001110110010100011000100000101000001101100100...
2866959C8EF2C4F5E08B77AA2C8008E3	0(0)	1010100011011101000110101111010111001100001000011011000110110101...
C49AE355075CD980479679D54782278E	0(0)	010001000010001010101011000011101111101001100110101101000111000...
80E3562EFOA98EB671C1EB8C0F6A072D	0(0)	000000000010001101100110001000010100010100110011110001110101001...
59627A0B1D557391A9E0DE046D08B61A	0(0)	11011001110110011110101101011011000101101011111000010101010111...
597C1522A4C028700130E5D7770EDD20	0(0)	110110011100011100011010100001000111011010111101010011010101010...
38467C62AC3D0A82E5C0A9A2B4845564	0(0)	1011101111111011111001011000100011101001100011110111000100001101...
81330704C08D9DD68C3A8B0D8E4151CC	0(0)	0000001100010001000100010100010000110000111011100101110001010111...
2D4D2E9992A151490849666910400ED9	0(0)	101010011110110101000100111101010010100101111000111000101010...
88DCCAA509CBEA8732934840532CBA2C	0(0)	000010110110011101000101000001110100010011000101011000100010001...
80EBA96602C4989E3CE4804C83C6793	0(0)	00000000000000010011011000001101101000111100010101001011010101...
60E88C9C3012FD27FD77FF1DCA71FFB7	0(0)	00000000000000011000100111000111010001110100010011111101100101...
80E291A8E3234DAF23453875155C4751	0(4)	0000000000100100011110000111001101011101100111111101101100011...
2F9F7620A1600C5A98B8967746CE1B5	0(0)	101011110010010011111001100001100111100110011011010011001110...
2F9A4CD7D167F90A89DF51D0878CB14	0(0)	10101111001000111001111011100010001001100111011110110101010011111...
5F1F782A0CA86C7E4AA9E08FA972BD7	0(0)	1101111110100100111101001000110011010100010100101101111010111101000...

Current Searches (0)

Number	Key	Type	Name	Status	Load	Packets Sent	Responses
4	1AD77E17776CA451651529C8EA9688F9	Search Keywords		Stopping	0 (0/0)	18/6	608
3	45F7CF06CBD795808476216A4E000180	Store File	Test.txt	Active	0 (0/0)	90/12	7
2	D63460691D7ACC4DC2625D819F9E3F52	Store Keyword	test	Active	52 (6/312)	28/10	6
1	D04D312FBA13176E85C7C2168C9009E8	Search Sources		Active	0 (0/0)	13/1	42
-1	146D85514AC08E3FF7C82948022E8FF8	Node Lookup		Stopping	0 (0/0)	8/0	0
-1	2905DD012C9C3CD94847968D463A891	Node Lookup		Active	0 (0/0)	2/0	0

Bootstrap

IP or Address: Port:

From known clients

Contacts

KAD: L'INTERFACCIA

Sulla sinistra: lista dei contatti presenti nella routing table. Per ogni contatto

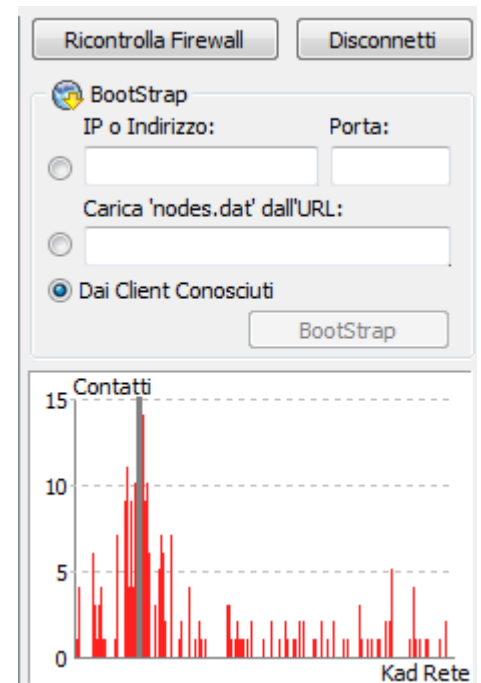
- una icona (diversi colori identificano contatti più nuovi/vecchi)
- identificatore del contatto
- permanenza del contatto in rete (codifica numerica della icona)
- tipo di client (versione di Emule)
- distanza

ID	Tipo	Distanza
4CCD95D5A988256DB197F55E837A049A	2(9)	011001110101010111001111001101110001100111100111101100010000010101110010000010011001001110100100...
4CC60A28906C2838077A2801484E4773	3(9)	011001110101111001010000110010100010000000000011011110001010000110001001110010001001110111101011...
4C39917A2781A9F9F2D650B760CBC908	2(9)	01100111101000011100101110011000100101111101110001111011001000100110001010010000011011001001101...
4C70A7AB1E575C00FF4ADA3BE0EBA125	3(8)	011001111110100011111101010010011010111000111000110010000110100000111100110101001011110011000001...
46B1F381E1E66081126403C833FB15A2	2(9)	011011010010100110101001011000110101000110001001111101001110100111010001111110100110010100110010...
46B4591D71B5996D7E0C164558FDEF64	2(9)	01101101001011000000001111111111110000011011010000011010000010110111011001001001110000101111111...
45D1295FB4E097339B59D6C433A2DAA2	2(8)	011011100100100101110011101111010000010010001111000000110101101101011000110001111011000000111110...
532002BC4914A9AF96DBB2ACE460A5EF	2(9)	011110001011100001011000010111101111100101111011001111011100011101010101010001011101010001010110...
525601DA11194CB267174D605EBD0539	2(9)	01111001110011100101101100111000101000010111011011011000110110101010010010010010101110011010...
51F5F29082EDDCDF1706E1AB46D01F9B	4(9)	011110100110110110101000011100100011001010000010010010001011011111010100100110001000011101010001...
50CDE0D6B30E6C9527937984E9E8A009	3(9)	01111011010101011011101000110100000000110110000111111000111111011110010000001101000111110111110...
57101B63B708E62C669EE327E1B5E170	2(9)	011111001000100001000001100000010000011101100111011100100100101010111101010011100100000001000010111011101...
54578A3129FA2ACFE2B9A9499BE9A5E7	3(9)	0111111110011111101000011010011100110011001010110111101010011100100001001001111100111110110011...
A88FC07B233E6F636BC3C2A666FF90FC	2(9)	100000110001011110011010100110011001001101010001111110110000101110101000010111011010010001011100...
AD514A8D8C962882FA591E3AABDD91FA	2(9)	1000011011001001000100000110111100111100111100110111001110101000111001110001110111100011000000...
AD516F0D54CA30684F159AF6B4CEF222	2(8)	1000011011001001001101011110111111001001001011010010000000001000110010001011111110000001100...
A0E6B8366C11BE1E19A9B4871065256E	3(7)	10001011011111101110001011010100110111000111111000101010011101101101010001101111101001001111101...
A00FD8B913409C2748DB37D3C71D3192	3(9)	1000101110010111100000100101101110100011001011110000100001001111100010110100010101010001001001...
4A00237A5FC000000000000000000000	2(8)	10001110010000001110001001011000111110000110010101011000011111001011101010010110001...






KAD: L'INTERFACCIA

Sulla destra

- informazioni per il bootstrap
- un grafico che mostra la distribuzione dei contatti nella routing table
 - asse delle x l'intero spazio degli identificatori
 - asse y numero di contatti in una specifica regione dello spazio degli identificatori
 - barra grigia: area con il maggior numero di contatti
 - identificatore del nodo vicino alla zona con la barra grigia



KAD: LE RICERCHE ATTIVE

	351	EDABCFCF5250A998DC4736F6514B55B9	Memorizza File	Computational Geometry algorithms and applicati...	In Arresto	0 (0 0)	37 13	10
	352	EE04B9ECAA7E7613124F9F8477897073	Memorizza Paro...	introduction	In Arresto	19 (10 193)	34 13	10
	353	2A8497A753C426BAF8C4E9A838809065	Memorizza File	Stochastic Processes - Sheldon M. Ross (1997).pdf	In Arresto	0 (0 0)	21 13	10
	354	365DA0DA7AFA1A620DC4BC8C8ACBBA29	Memorizza File	KAD 101 Everything you want to know on KAD - ...	Attiva	0 (0 0)	26 12	9
	355	57DC12177D7F33E95023E23D318CC3F7	Memorizza Paro...	probability	Attiva	1 (7 13)	17 9	7

Per ogni ricerca attiva:

- Tipo di ricerca (icone mostrate nei lucidi precedenti)
- Target ID
- Nome del file/parola chiave ricercati
- Stato della ricerca (attiva/stoppata)
 - stoppata: si è ottenuto un numero sufficiente di risposte e quindi la ricerca è stata momentaneamente sospesa

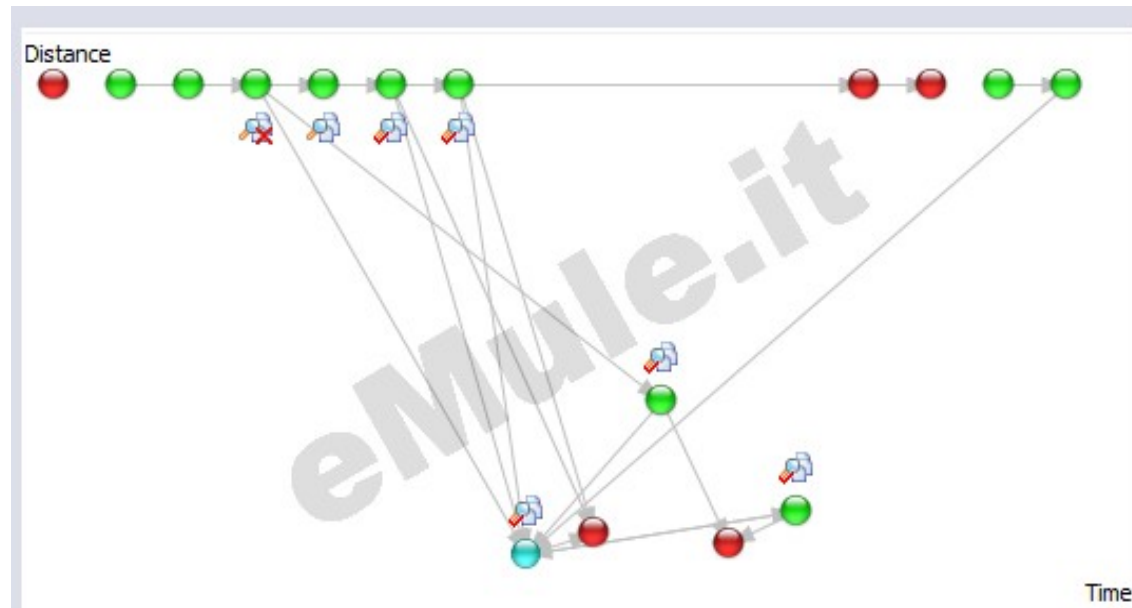
KAD: LE RICERCHE ATTIVE

351	EDABCFCF5250A998DC4736F6514B55B9	Memorizza File	Computational Geometry algorithms and applicati...	In Arresto	0 (0 0)	37 13	10
352	EE04B9ECAA7E7613124F9F8477897073	Memorizza Paro...	introduction	In Arresto	19 (10 193)	34 13	10
353	2A8497A753C426BAF8C4E9A838809065	Memorizza File	Stochastic Processes - Sheldon M. Ross (1997).pdf	In Arresto	0 (0 0)	21 13	10
354	365DA0DA7AFA1A620DC4BC8C8ACBBA29	Memorizza File	KAD 101 Everything you want to know on KAD - ...	Attiva	0 (0 0)	26 12	9
355	57DC12177D7F33E95023E23D318CC3F7	Memorizza Paro...	probability	Attiva	1 (7 13)	17 9	7

- Load: popolarità di una parola chiave pubblicata
 - nel primo campo la media dei valori di popolarità restituiti dalle ricerche, se maggiore di 20 la parola chiave è considerata molto popolare e non viene più pubblicata per un certo intervallo di tempo
 - secondo campo(due numeri tra parentesi):
 - numero di nodi da cui si sono avute risposte
 - somma dei load di questi nodi
- Pacchetti inviati/risposte: statistiche sulle ricerche

KAD: L'INTERFACCIA

- Esiste anche una interfaccia che consente di visualizzare in tempo reale informazioni su una ricerca specifica
- Ogni pallino rappresenta un nodo
- I nodi posizionati in basso sono quelli più vicini al targetID
- Freccia dal nodo a al nodo b indica che a ha fornito informazione su b
- A seconda del tipo di risposta restituito da un nodo un colore diverso per il nodo stesso



KAD: L'INTERFACCIA

eMule v0.50a

Disconnetti Kad Server Trasferime... Cerca File Condi... Messaggi IRC Statistiche Opzioni Strumenti Aiuto

Dettagli di Ricer...

Distanza

Ricontrolla Firewall Disconnetti

Bootstrap

IP o Indirizzo: Porta:

Carica 'nodes.dat' dall'URL:

Dai Client Conosciuti

Bootstrap

Contatti

Tempo Kad Rete

Ricerche Correnti (5)

Numero	Chiave	Tipo	Nome	Stato	Carico	Pacchetti Inviati	Risposte /
-1	2B85515C0ED14CA837D3C7534C451D3F	Analisi Nodo		In Arresto	0 (0 0)	1 0	1
-1	2B89CFD674ED6D44EF319C00E150CCB2	Analisi Nodo		In Arresto	0 (0 0)	1 0	1
653	F13DB08C2399798E77FBBA72A08DCA12	Memorizza File	Introduction to Time Series and Forecasting, 2ed...	Attiva	0 (0 0)	8 3	2
...