Lezione n.10

Freenet
Materiale didattico: articoli distribuiti a lezione
Laura Ricci

Il Progetto Freenet

- Progetto Freenet: prende spunto da un progetto sviluppato nel 1999 da Ian Clarke, uno studente undergraduate (laurea triennale) di Edinburgo

- Contemporaneo di Napster, Gnutella. Svantaggio principale di questi sistemi: non garantiscono l'anonimato
  - È possibile individuare l'host da cui si sta scaricando materiale
  - È possibile individuare chi ha inviato una query

- Obiettivo fondamentale: garantire l'anonimato, rendere più ardua la censura del materiale condiviso

- Obiettivi secondari:
  - disponibilità dell'informazione
  - affidabiltà
  - efficienza
  - scalabilità

Il Progetto Freenet

- Luglio 1999: pubblicazione di un primo articolo contenente le idee fondamentali alla base del sistema
- Successivamente: Clarke ed un gruppo di volontari iniziano a lavorare al progetto. I membri del gruppo di lavoro appartengono a diversi paesi
- Sviluppo di Freenet avviene in modo decentralizzato
- Marzo 2000: versione 0.1 di Freenet
- 2000: il lavoro di Clarke "Freenet: A Distributed Anonymous Information Storage and Retrieval System" risulta il lavoro più citato nel campo della computer science, nel 2000 (fonte Citeseer)

Freenet vs. Napster, Gnutella

- Napster, Gnutella: forniscono un servizio di condivisione di files.

  i files sono pubblicati sulla memoria dell'host locale

  i files rimangono disponibili sulla rete solo durante il periodo in cui l'utente rimane on-line

- Freenet: fornisce un servizio di memorizzazione e ricerca distribuita di files

  ogni nodo mette a disposizione uno spazio di memorizzazione

  l'informazione pubblicata viene memorizzata su un insieme di nodi della overlay network

  l'informazione rimane sulla rete anche quando l'host che l'ha pubblicata si disconnette

  rete Freenet = file system distribuito

Freenet: Caratteristiche Generali

- ogni nodo Freenet fornisce uno spazio di memorizzazione
- ad ogni file inserito nella rete viene associata una chiave GUID (Globally Unique Identifier), calcolata mediante consistent hashing (SHA-1)
- il file viene inizialmente memorizzato su un insieme di nodi della rete
- un file può successivamente essere replicato su altri nodi
- routing

  guidato dalla conoscenza della chiave del file da ricercare

  adattivo: le tabelle di routing vengono modificate durante la ricerca delle chiavi

Freenet: Caratteristiche Generali

- Ogni nodo memorizza
  - un insieme di files
  - una tabella di routing che contiene
    - riferimenti ad altri nodi
    - alcune chiavi che individuano files memorizzati su tali nodi

- i riferimenti contenuti nelle routing tables definiscono la overlay network

- Routing:
  - basato su steepest descendant hill-climbing serch
  - associa HTL (Hops to Live) alle queries per limitarne la diffusione
    - L'HTL può essere incrementato, fino ad un valore massimo
  - associa identificatori casuali di 64 bits alle queries, per evitare loops
    - ogni nodo memorizza una lista degli identificatori delle queries già inoltrate

Freenet: Pubblicazione di Dati Condivisi

Per pubblicare un file F su una rete Freenet, un nodo:

- assegna un una chiave K ad F
- controlla se la chiave generata è già esistente sul nodo locale. In tal caso restituisce un messaggio di collisione.
- se la chiave non esiste

  analizza le chiavi memorizzate nella propria tabella di routing

  invia un messaggio *insert(K,HTL)* al nodo che possiede la chiave il cui valore numerico si avvicina maggiormente a K.

  il valore del HTL (Hops to Live) indica il numero di copie di D che devono essere memorizzate nella rete

Freenet: Pubblicazione di Dati Condivisi

quando un nodo riceve un messaggio *insert(K,HTL)*

- controlla se possiede K, in questo caso restituisce un messaggio di collisione. Il messaggio viene propagato indietro fino al nodo sorgente.

- in caso contrario decrementa HTL

  se il valore di HTL è diverso da 0, inoltra K al vicino che possiede la chiave numericamente più vicina a K.

  se il valore di HTL è uguale a 0, invia indietro lungo il percorso individuato dal messaggio di insert un messaggio "all clear". Questo messaggio indica che non si sono verificate collisioni lungo il cammino percorso.

- se non si è verificata collisione, il messaggio viene memorizzato in ogni nodo lungo il percorso individuato dal messaggio di *insert(K,HTL)*

Freenet: Routing

Algoritmo di Routing: quando N riceve la chiave K

- se N possiede il file corrispondente a K, la query non viene inoltrata

  N invia il file al nodo che lo aveva richiesto.

  il file viene propagato indietro lungo il cammino P percorso dalla query per raggiungere N,

  al file viene associato un tag che identifica N

- se N non possiede il file

  ricerca nella routing table la chiave K' numericamente più vicina a K

  se K' esiste, inoltra K al nodo N' associato a K'

  attende una risposta da N'

Freenet: Routing

Quando un nodo riceve una risposta ad una query da un nodo vicino

- se il vicino restituisce un messaggio di fallimento, request failed, cerca la chiave numericamente più vicina a K, scelta tra le rimanenti

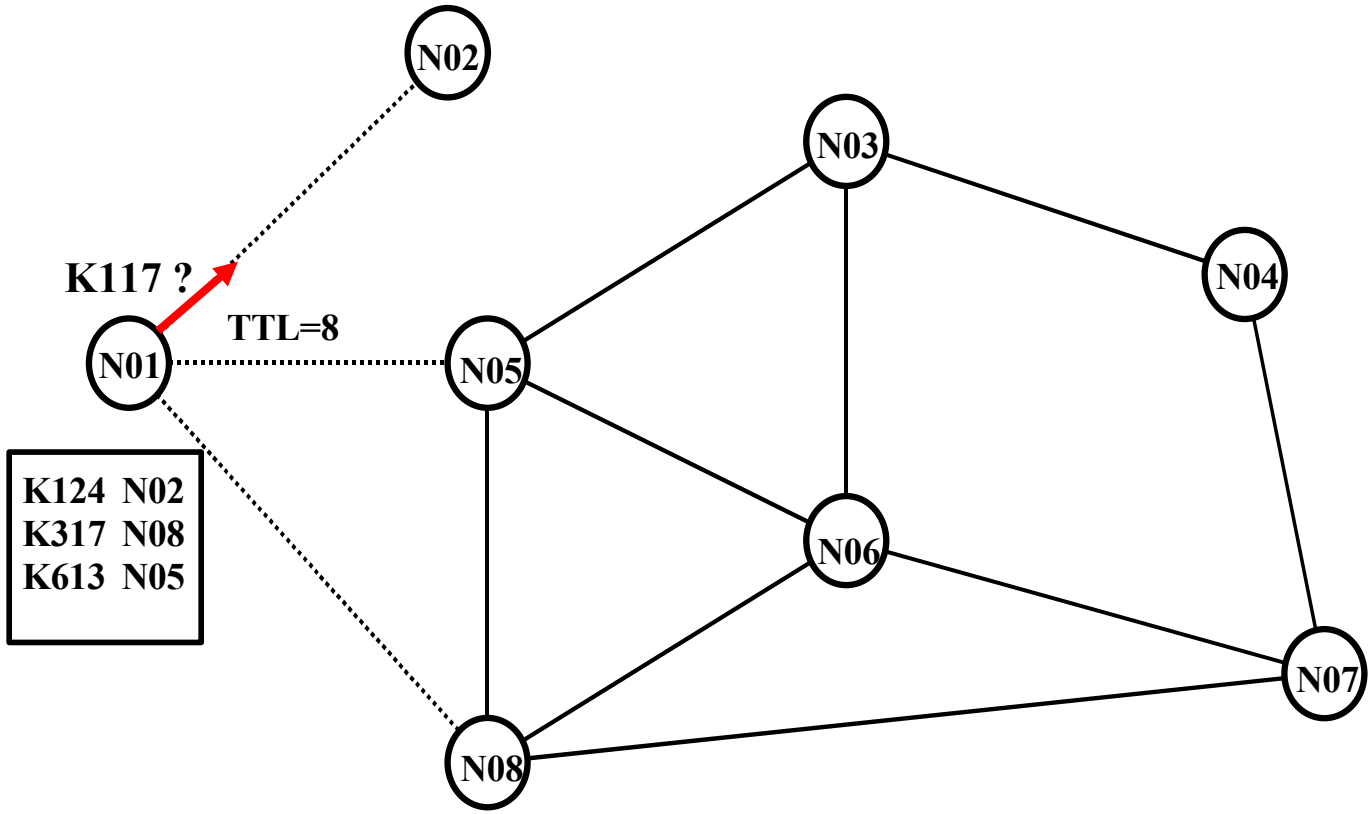
  quando tutti i vicini sono stati contattati senza successo, restituisce un messaggio request failed al vicino che gli aveva inviato la query

- se il vicino invia una risposta positiva (K,P) ad una query precedente

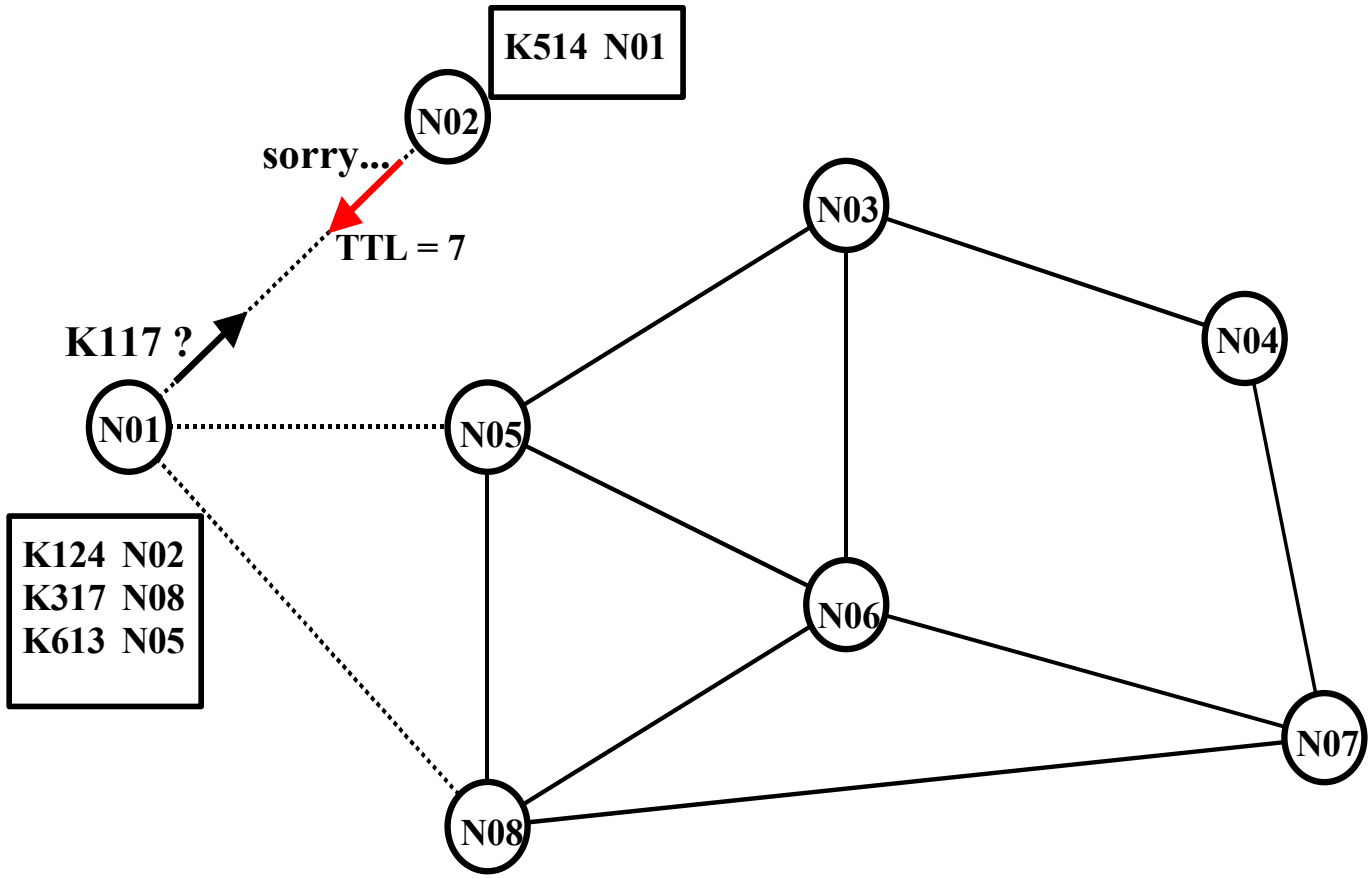
  aggiunge una nuova entrata (K,P) alla propria routing table

  può memorizzare una copia del file nella memoria locale (dipende dalla sua distanza dal nodo P)

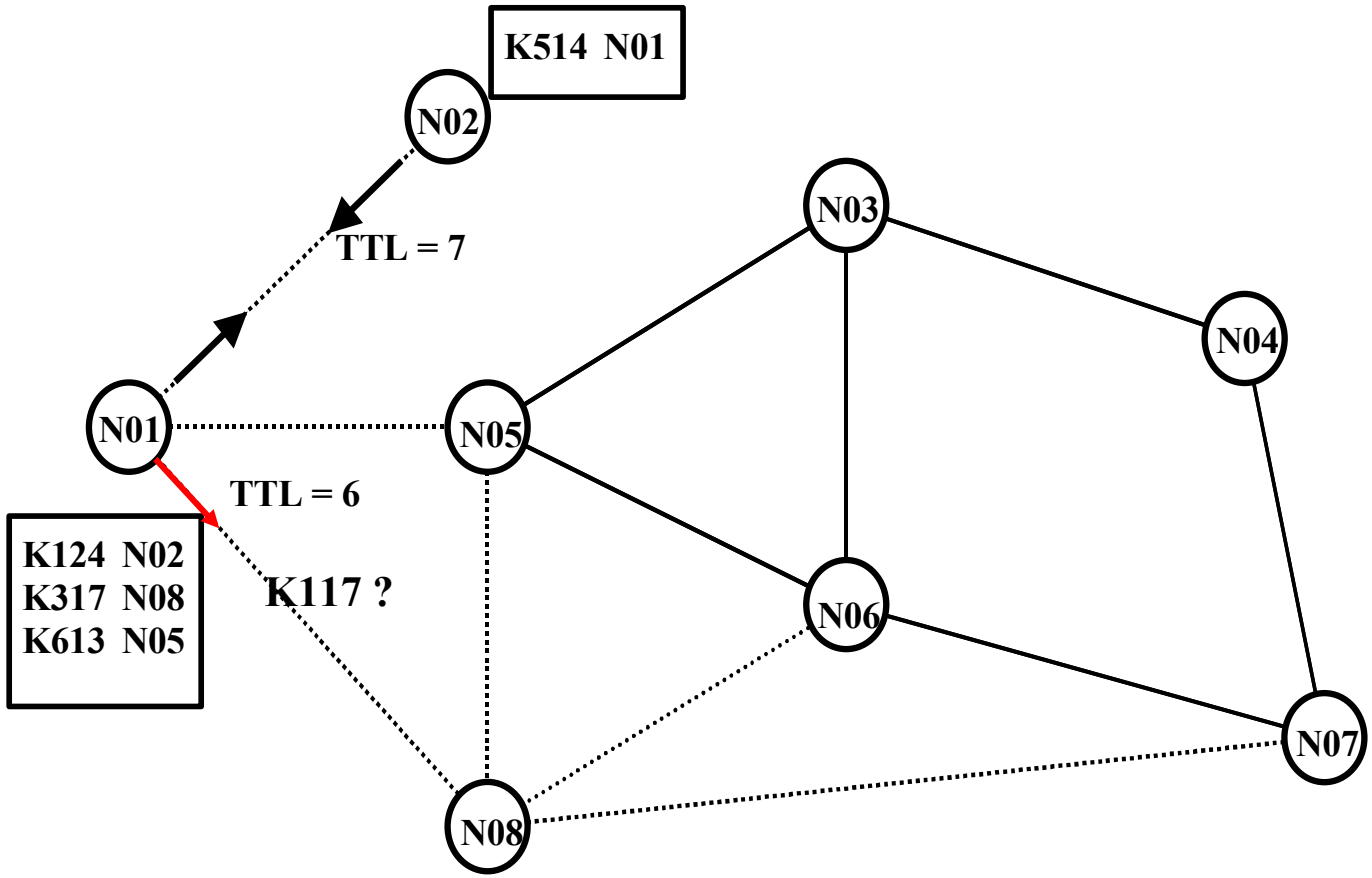
Freenet Routing



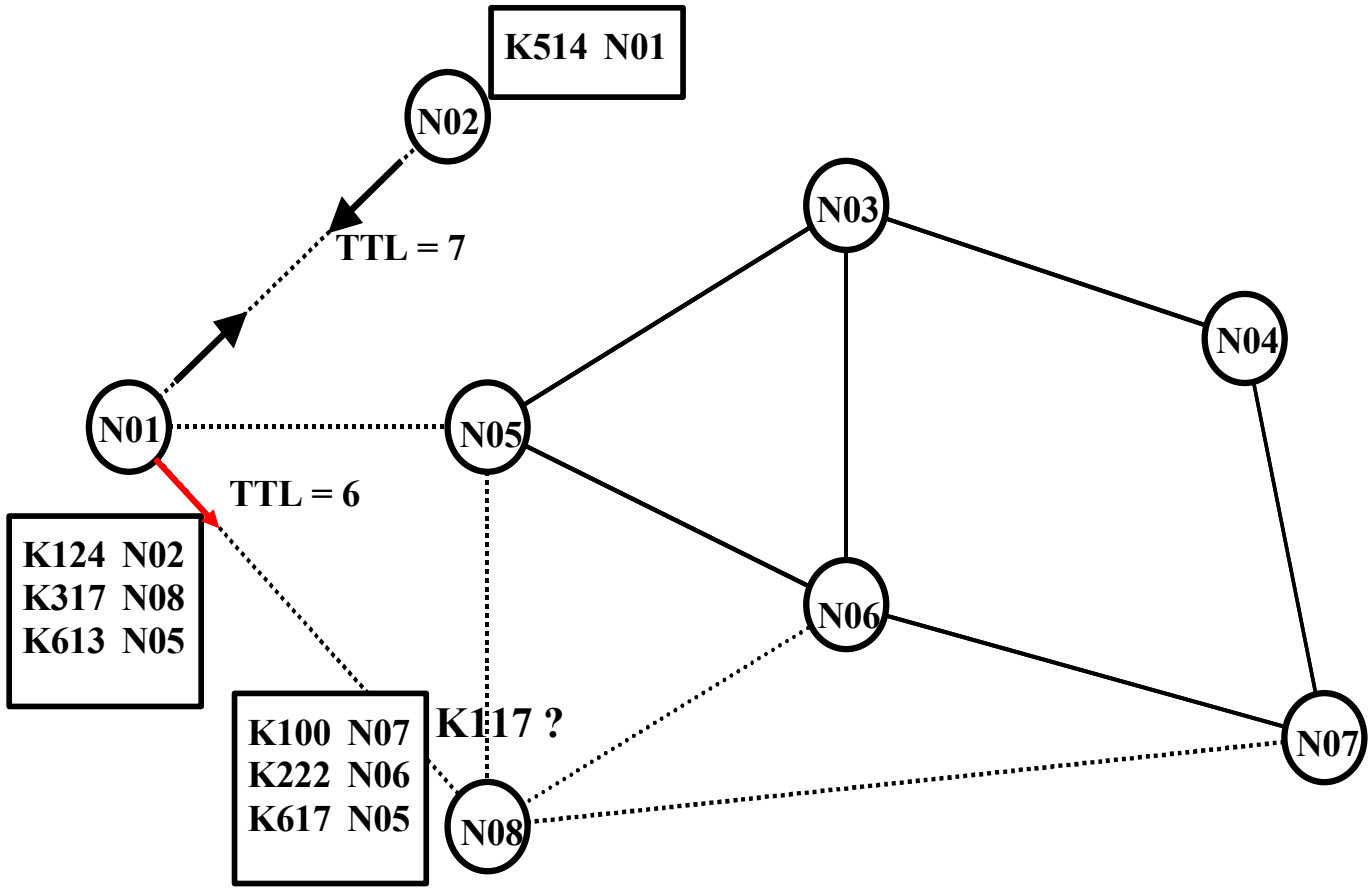
Freenet Routing



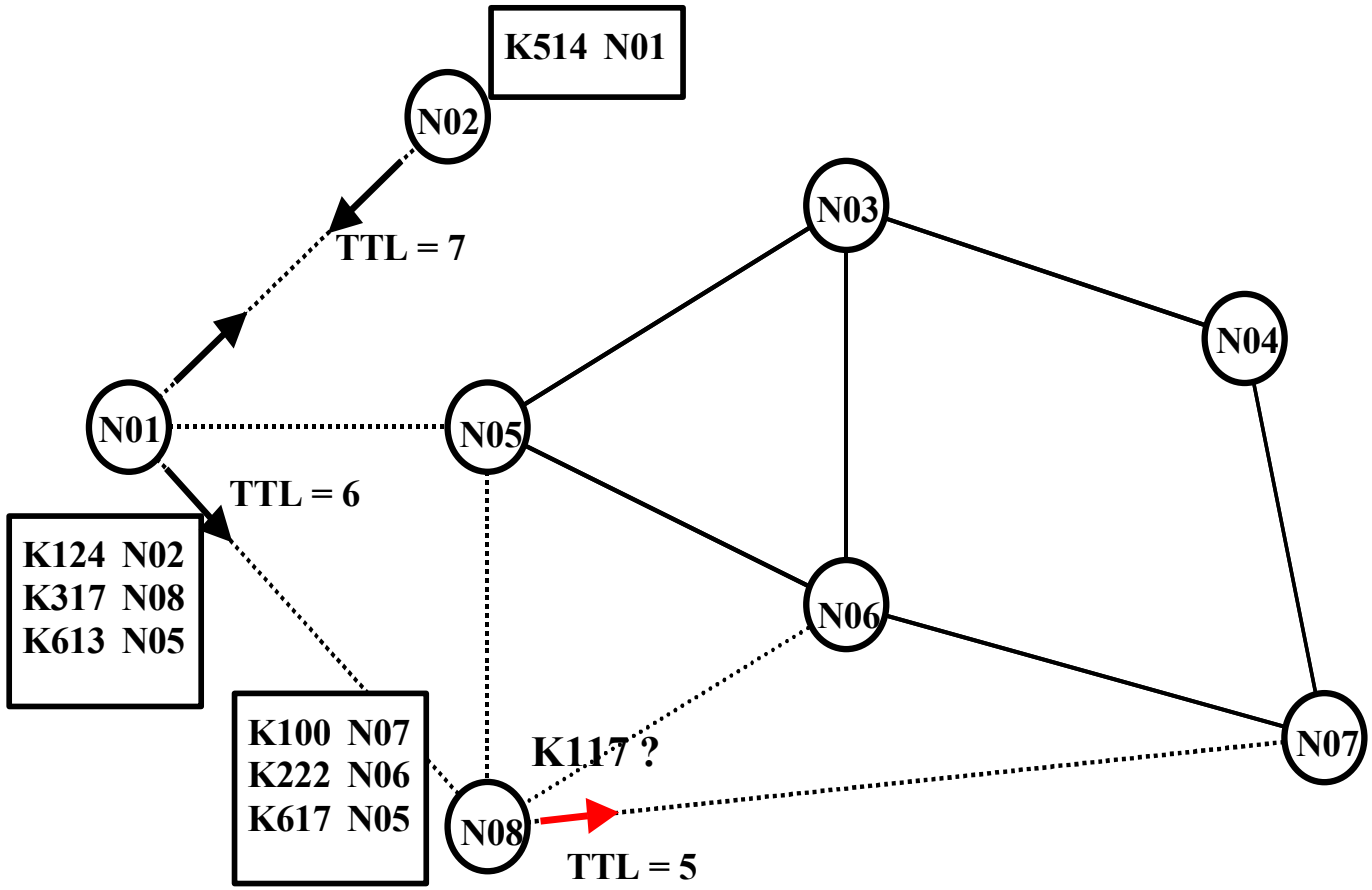
Freenet Routing



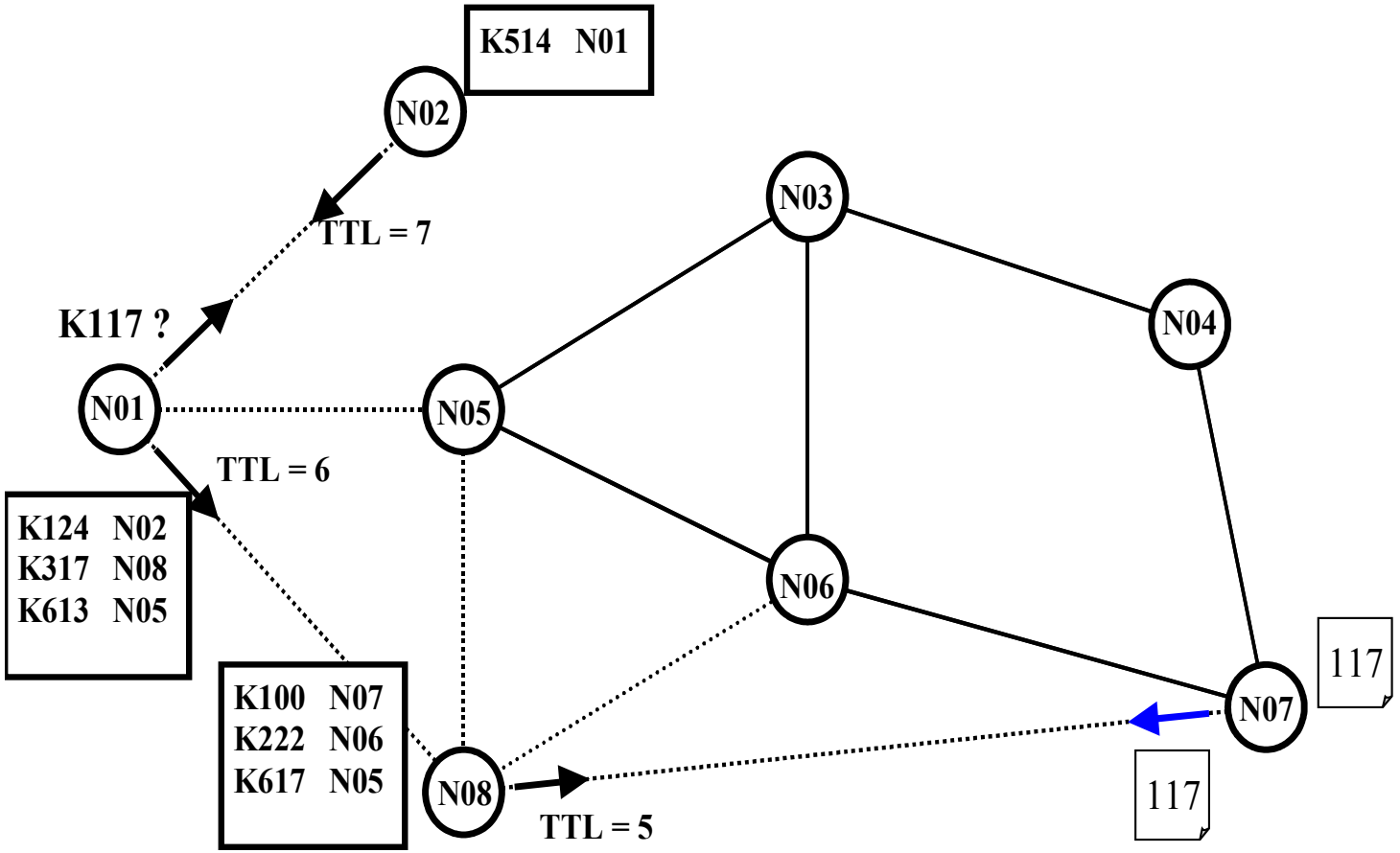
Freenet Routing



Freenet Routing

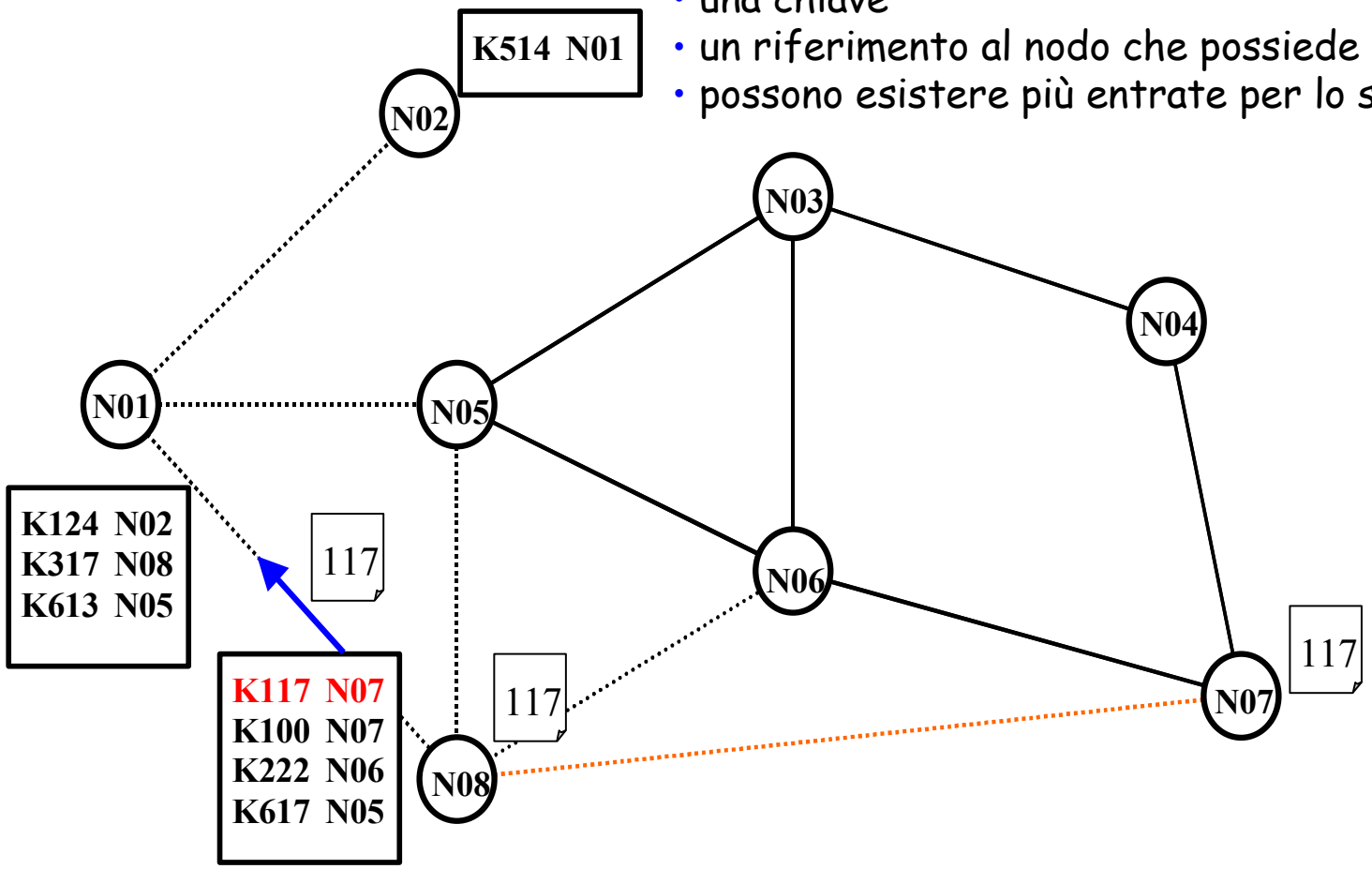


Freenet Routing



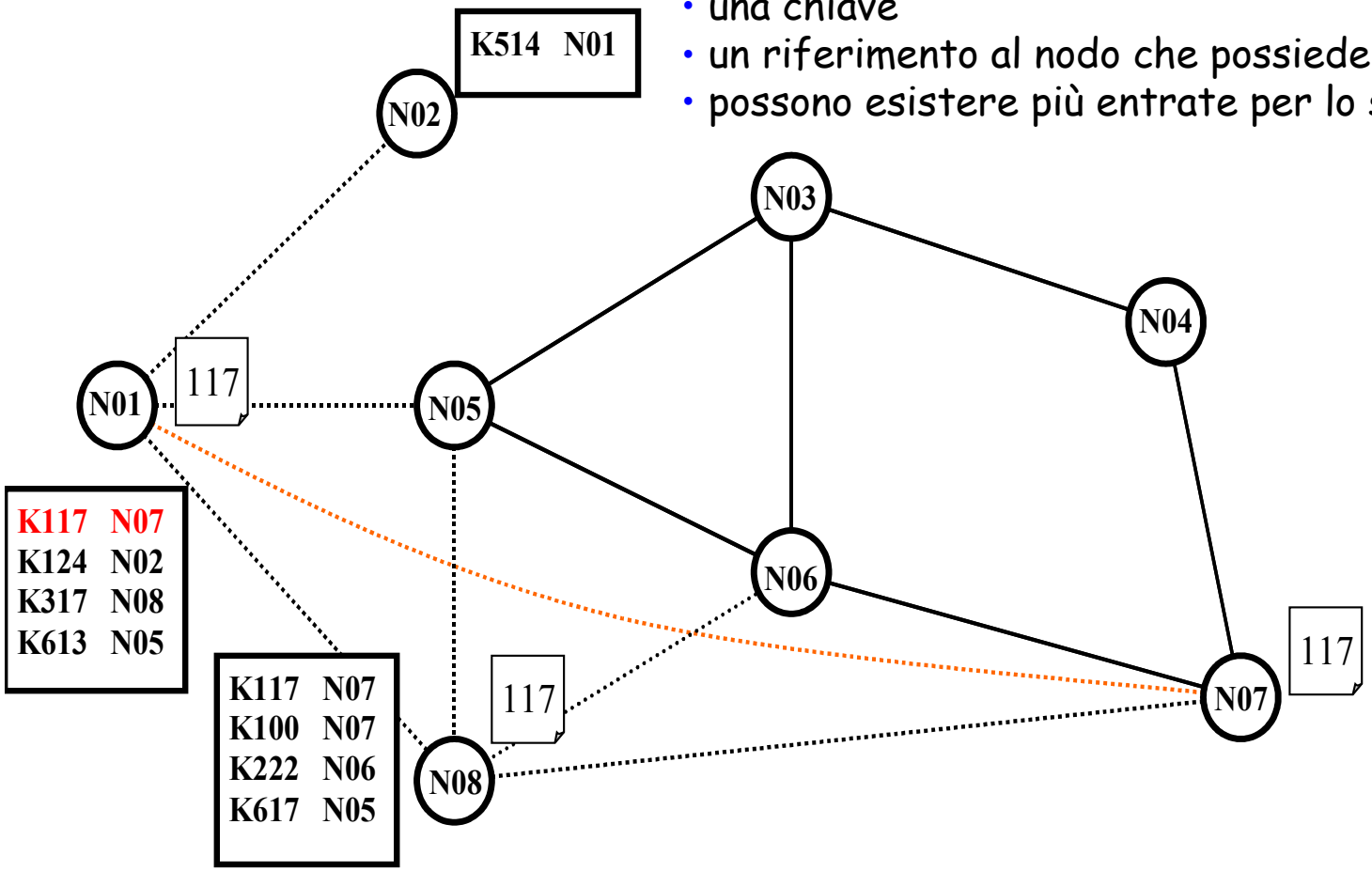
Freenet Routing

Routing Table: ogni entrata contiene
- una chiave
- un riferimento al nodo che possiede la chiave
- possono esistere più entrate per lo stesso nodo



Freenet Routing

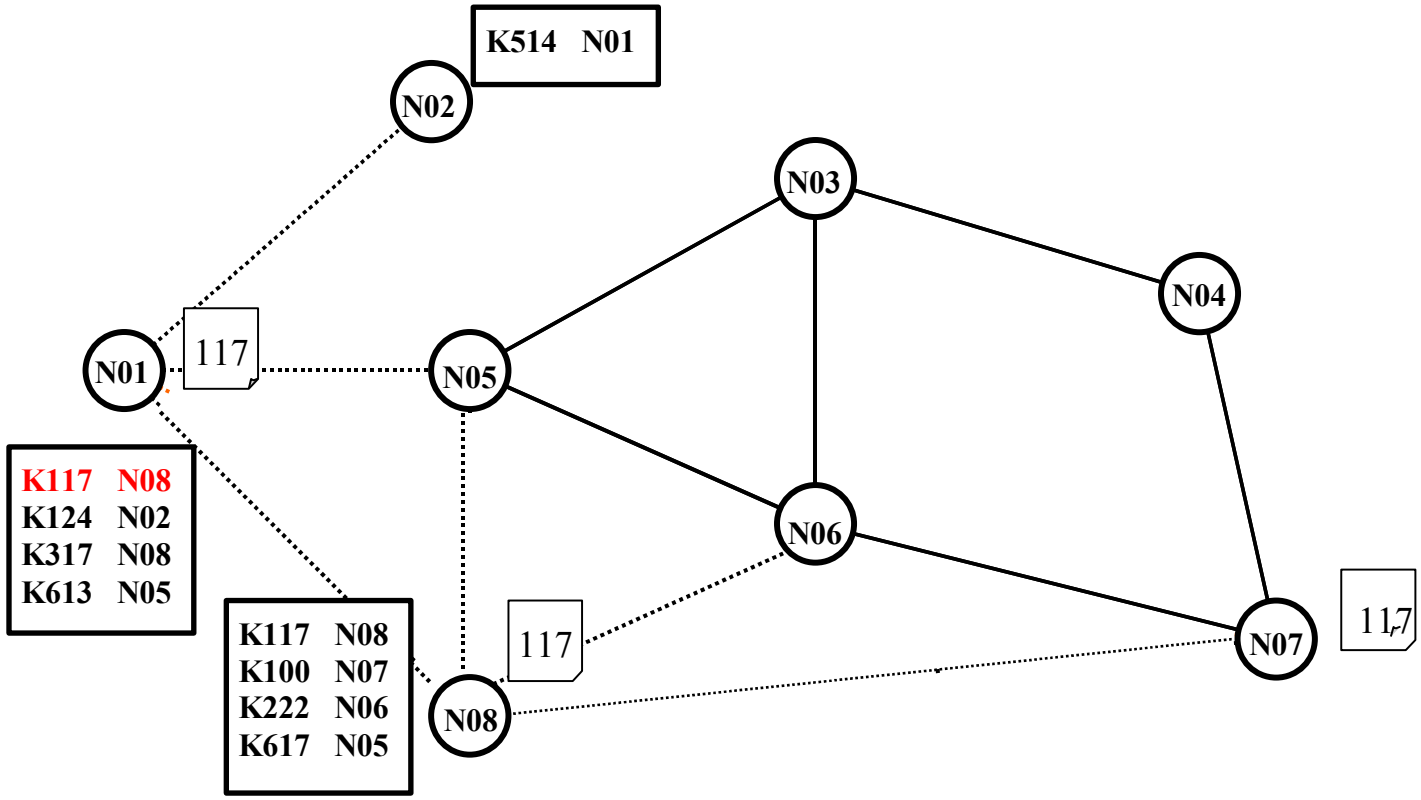
Routing Table: ogni entrata contiene
- una chiave
- un riferimento al nodo che possiede la chiave
- possono esistere più entrate per lo stesso nodo



Freenet Routing

- I nodi lungo il cammino possono decidere di memorizzare una copia del file

Esempio: la chiave 117 viene memorizzata da N08, N01 inserisce un riferimento a N08



Freenet: Request Failed Messages

Request failed messages: vengono inviati quando

- un nodo non possiede la chiave K e

  non possiede riferimenti a nodi vicini

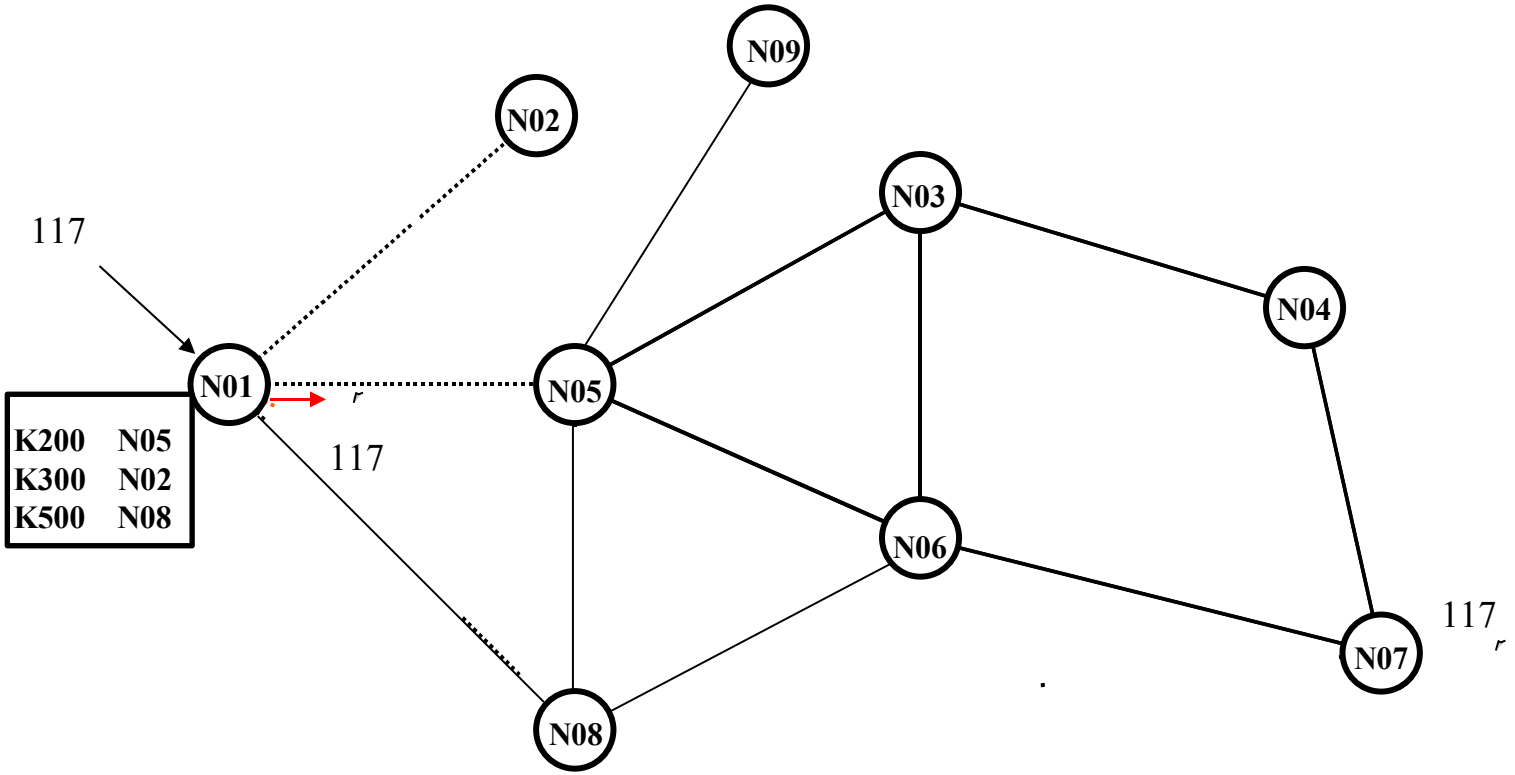
  oppure

  i vicini contattati non posseggono K

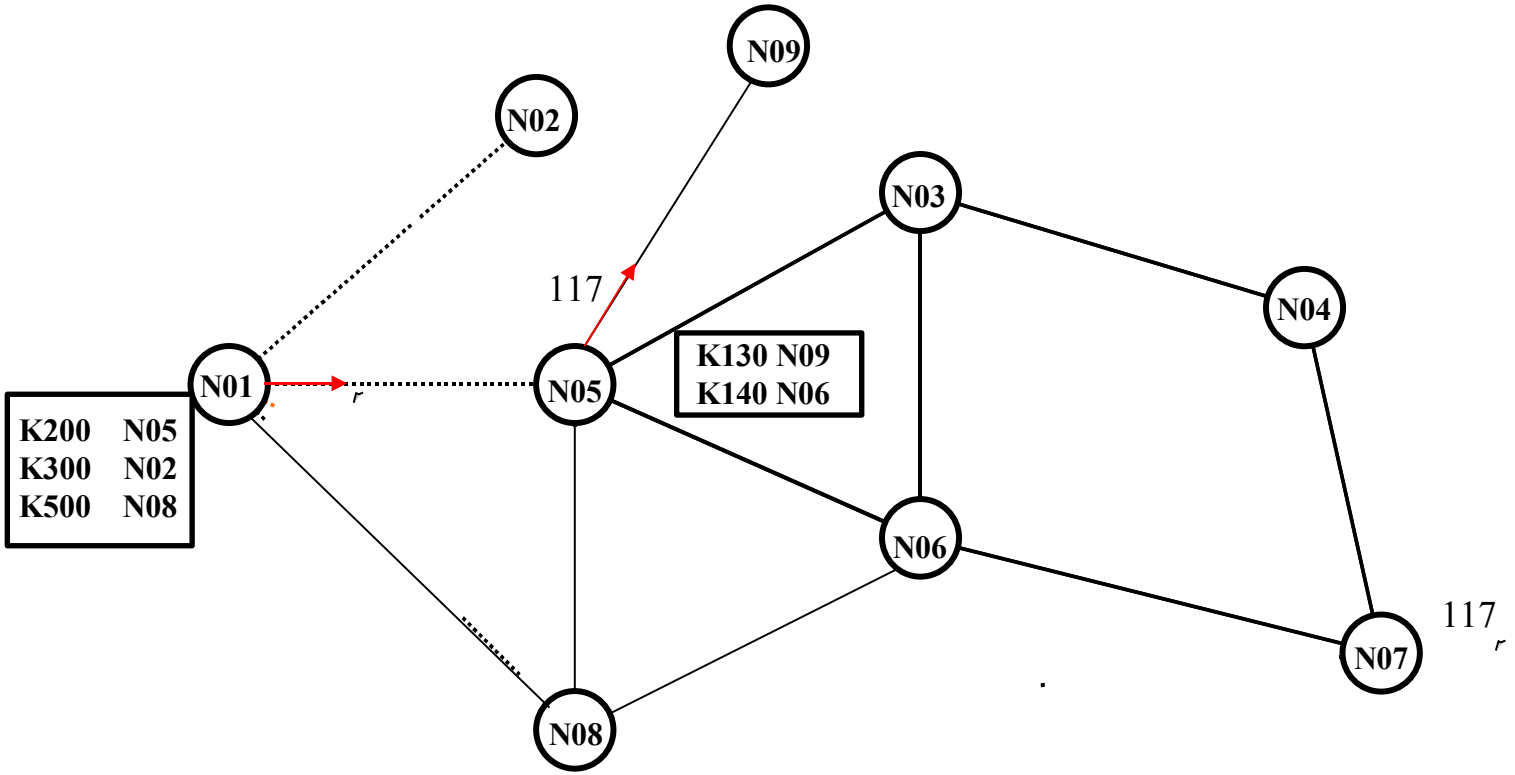
- un nodo individua un ciclo nel routing: il nodo ha già inoltrato in precedenza k

Quando un nodo riceve un messaggio di request failed, invia K al vicino che possiede la chiave numericamente più vicina a K, scelta tra le chiavi rimanenti

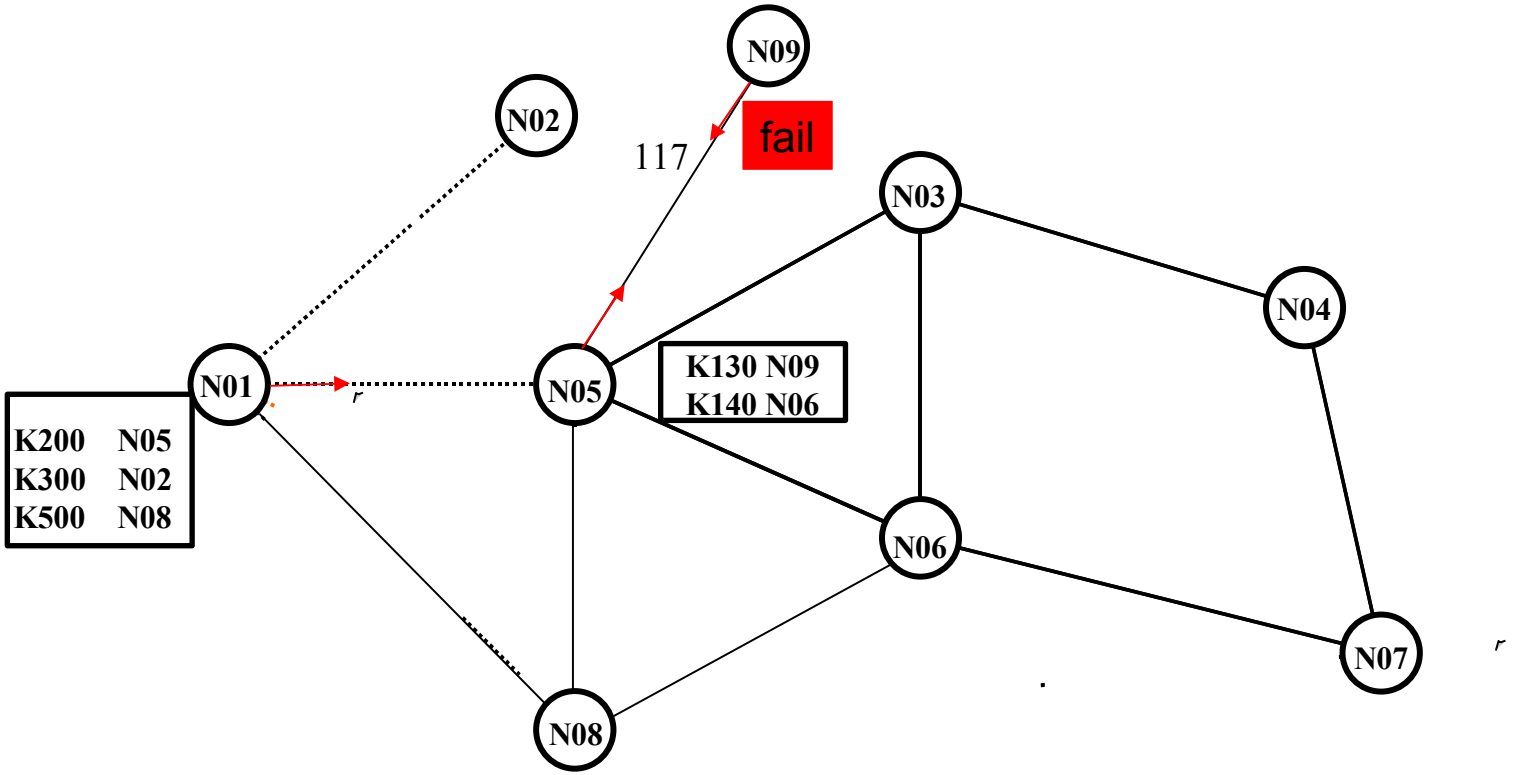
Freenet Routing



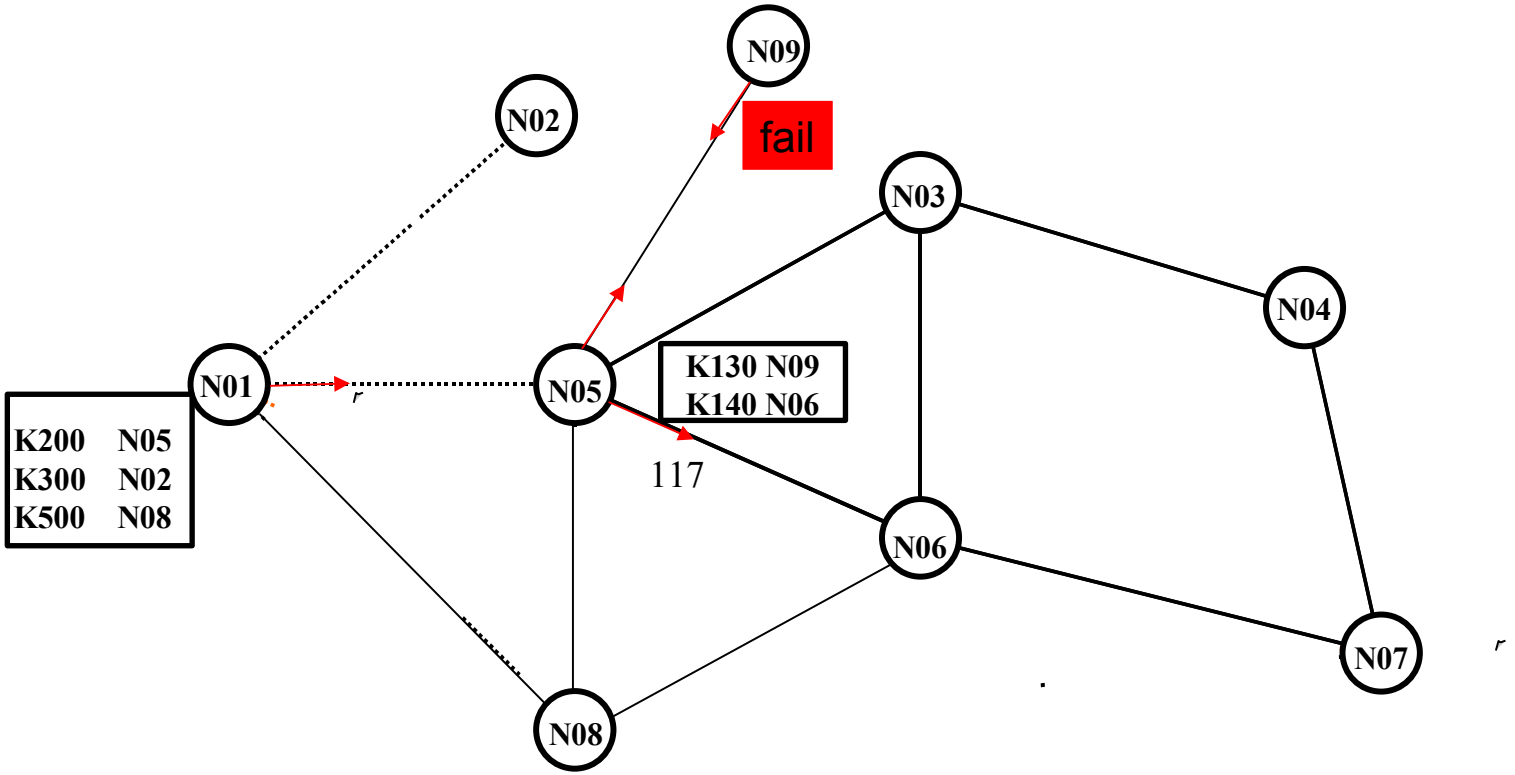
Freenet Routing



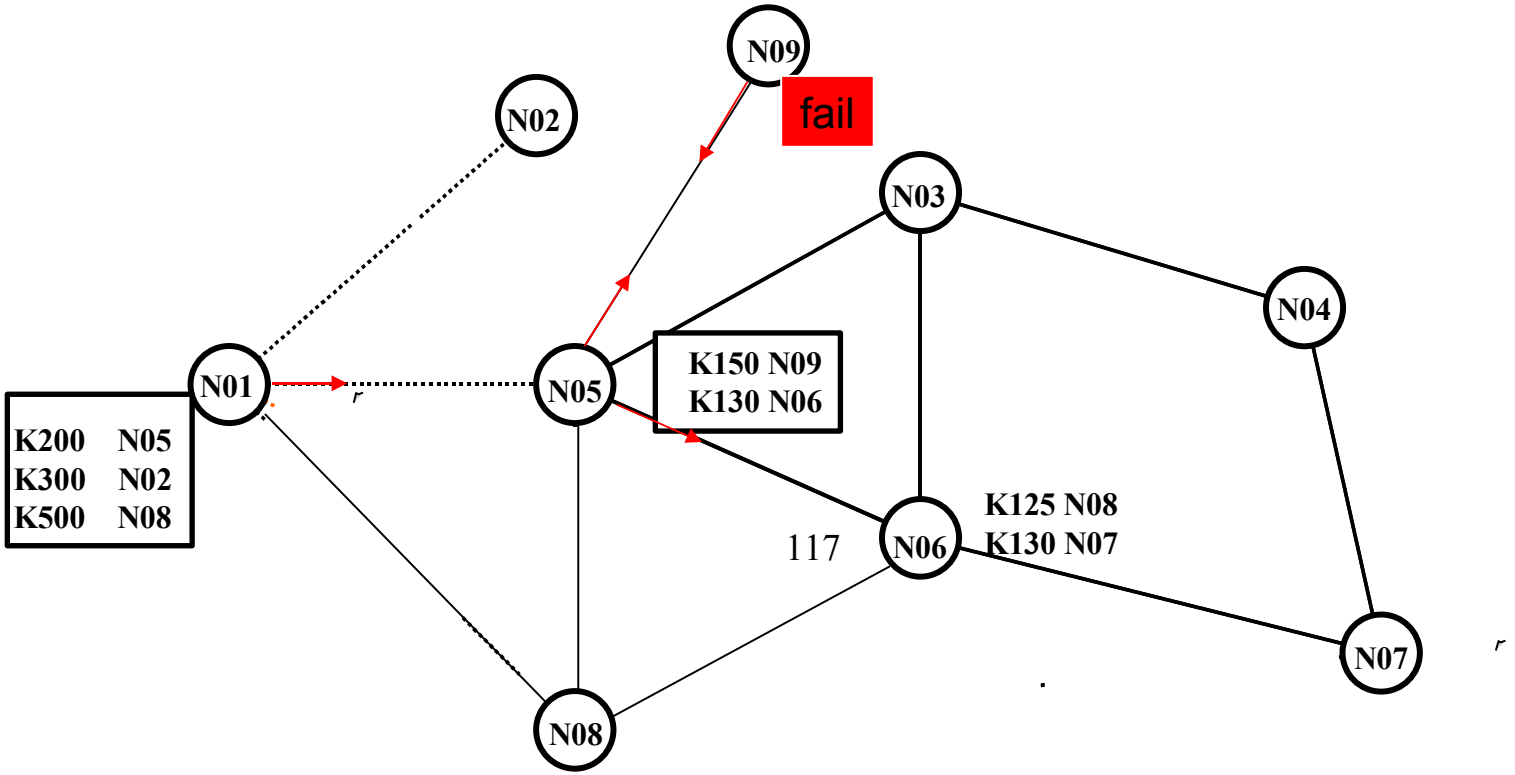
Freenet Routing



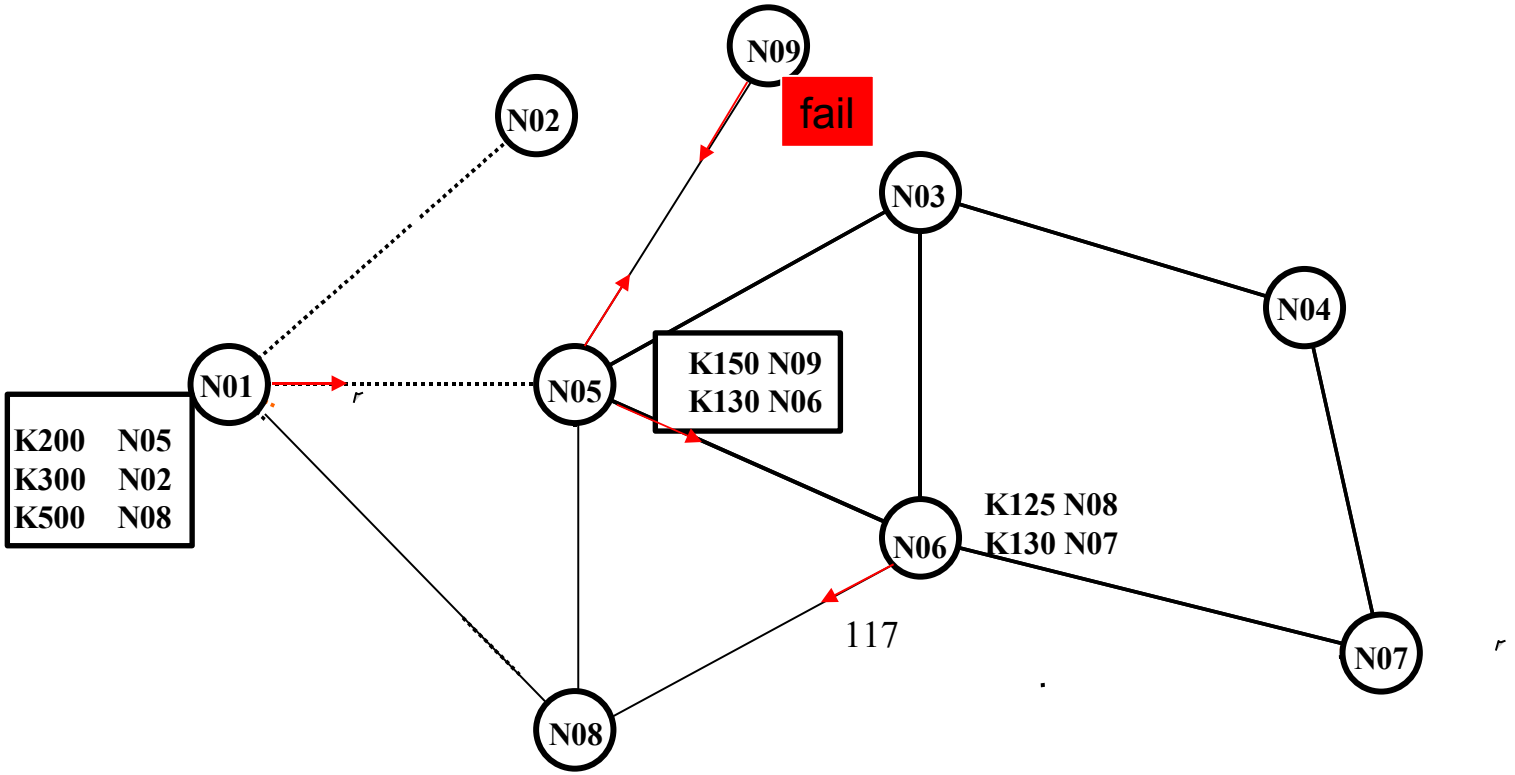
Freenet Routing



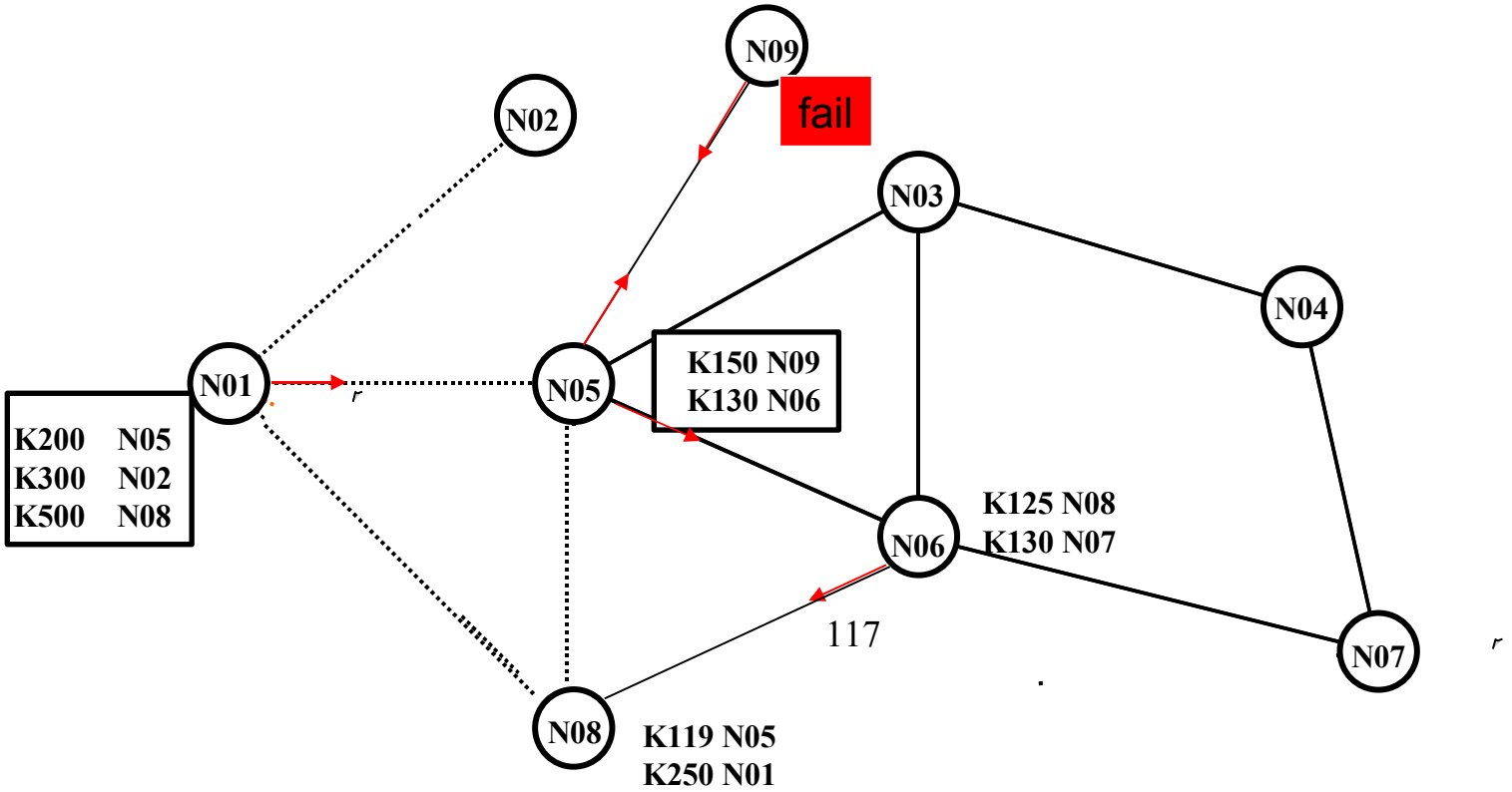
Freenet Routing



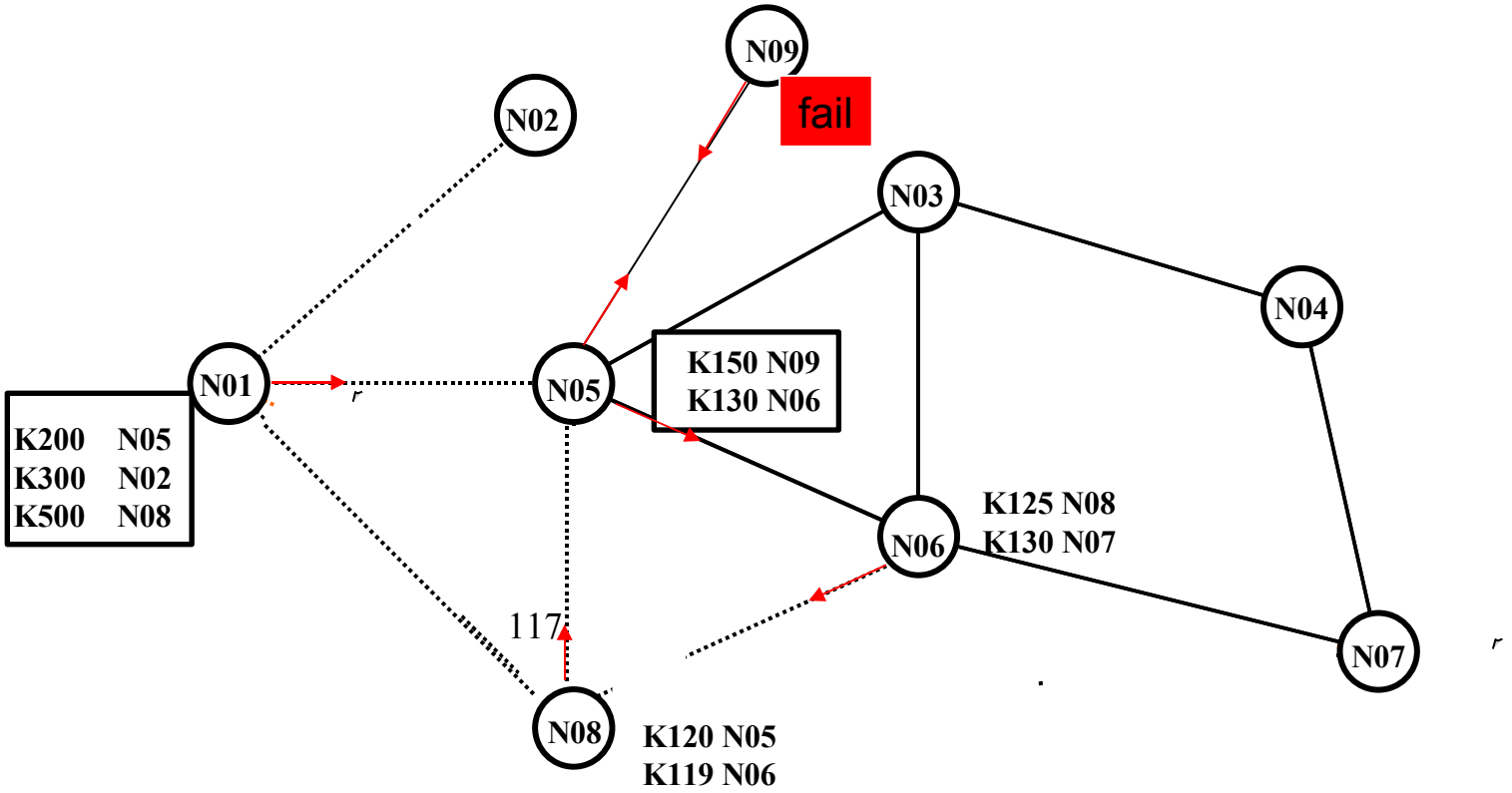
Freenet Routing



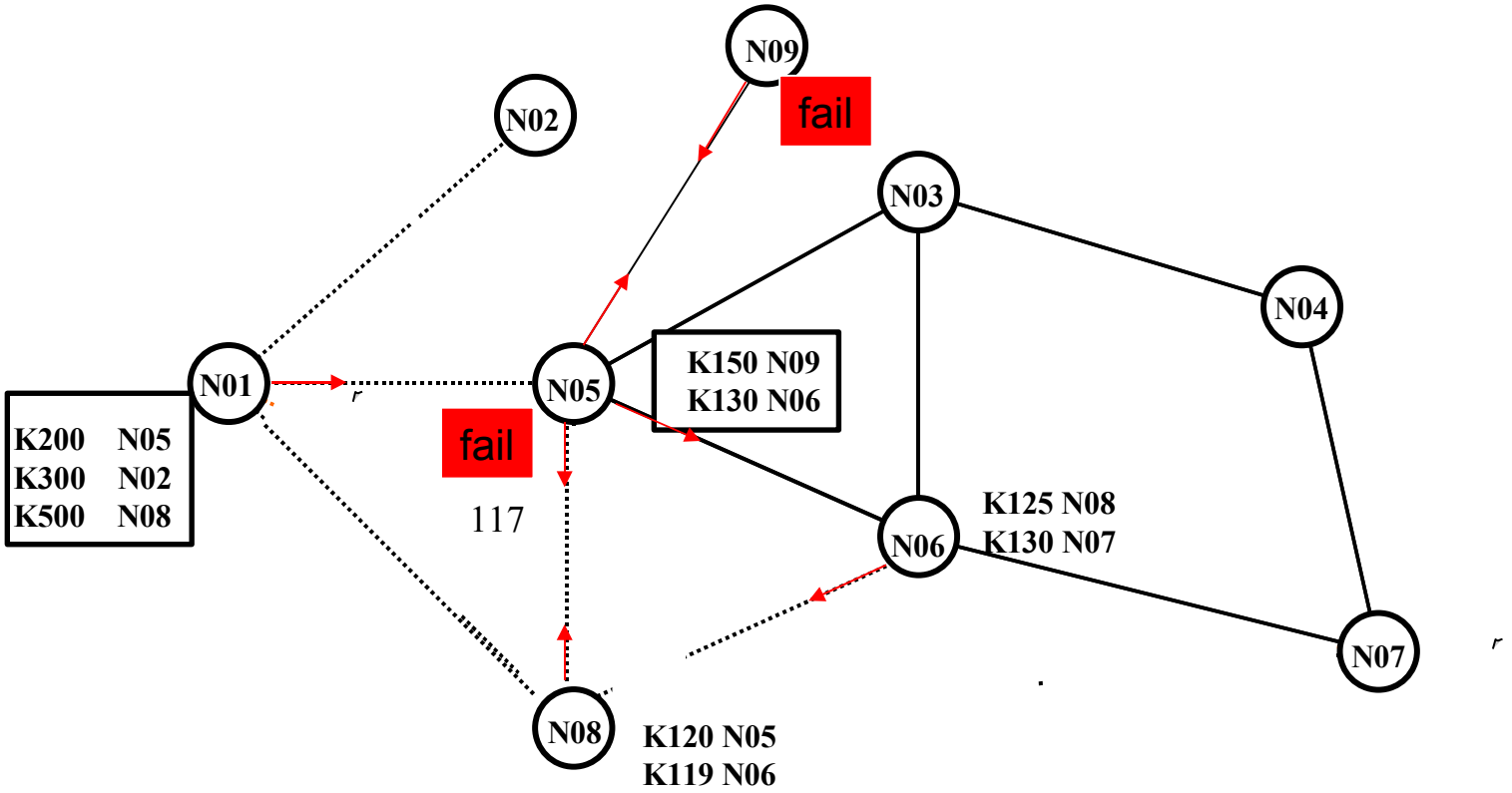
Freenet Routing



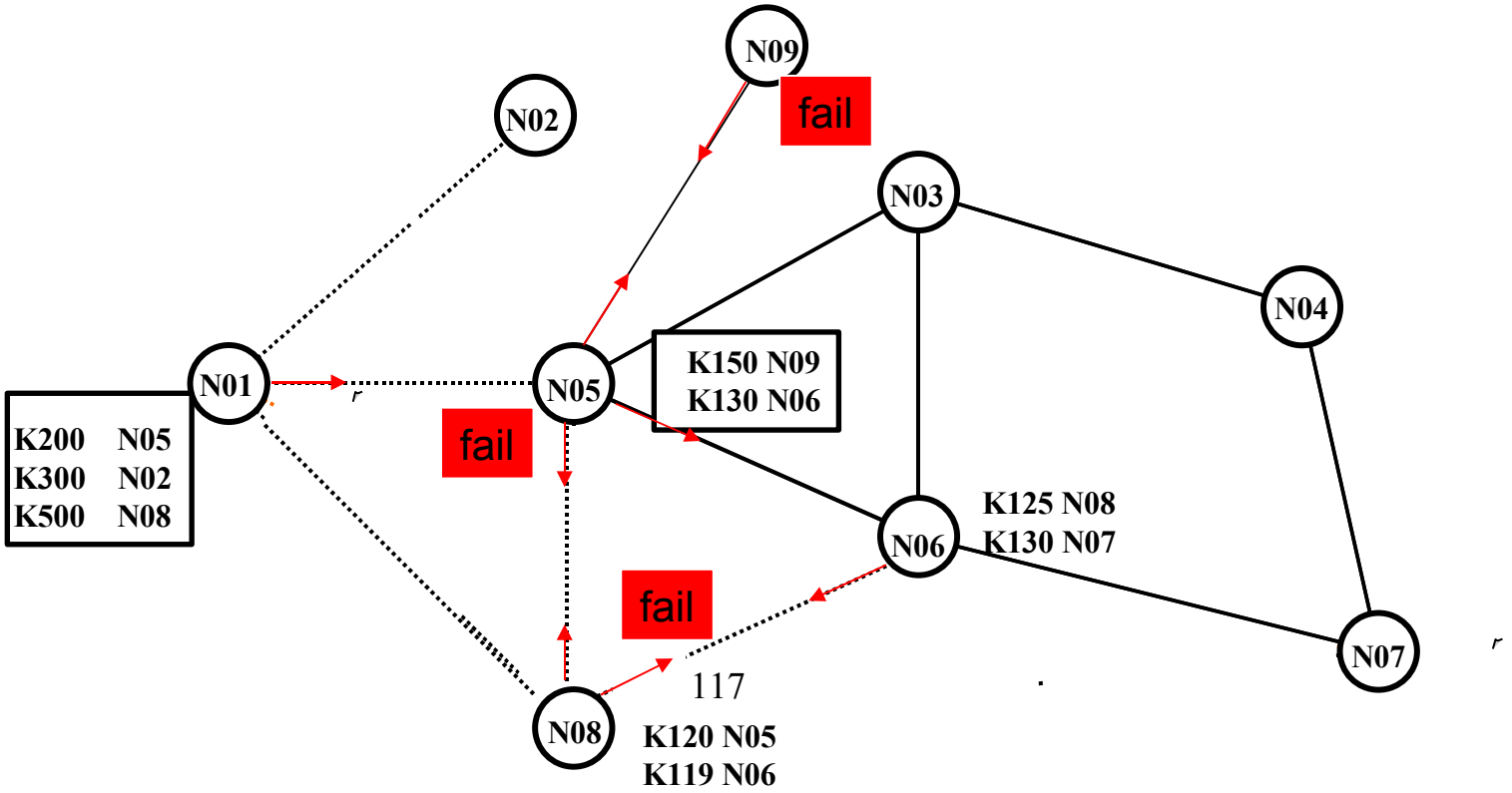
Freenet Routing



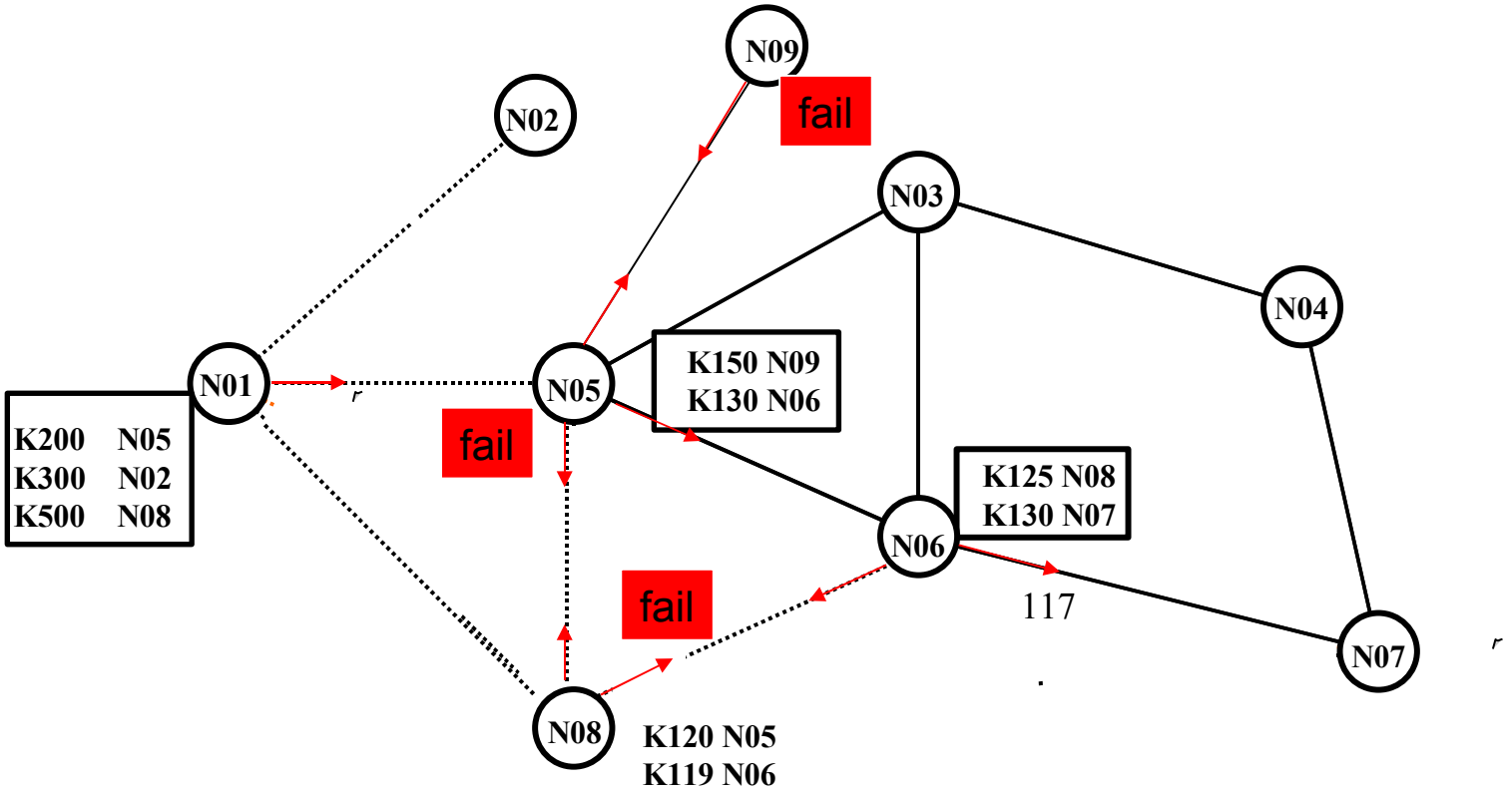
Freenet Routing



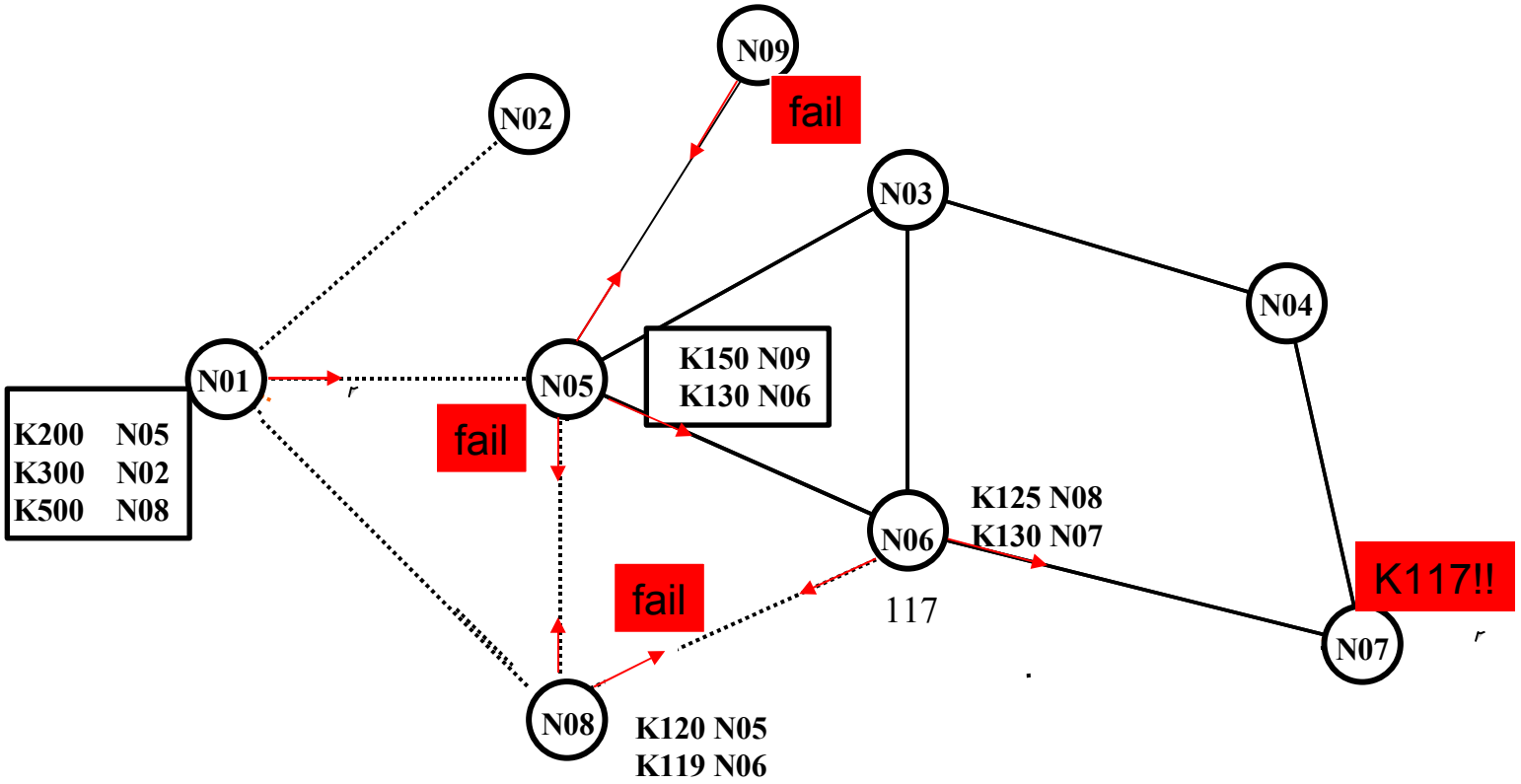
Freenet Routing



Freenet Routing



Freenet Routing



Freenet vs. DHT

- Distributed Hash Tables:

  ogni chiave viene associata in modo deterministico ad uno ed un solo nodo della rete (esempio: il nodo successore sull'anello, per Chord). Questo consente di limitare la complessità della ricerca.

  la replicazione delle chiavi è introdotta solo per aumentare l'affidabilità del sistema.

  non producono falsi negativi, garantiscono limiti di complessità per algoritmi di ricerca

- Freenet:

  ogni chiave viene allocata su un insieme di nodi. L'algoritmo di routing tende a clusterizzare chiavi simili sullo stesso nodo.

  può produrre falsi negativi, complessità di ricerca limitata mediante HTL.

  obiettivo del progetto: garanzia di anonimato dei nodi più che efficienza di algoritmi di ricerca

Freenet: Garantire l'Anonimato

- Obiettivo; garantire l'anonimato di chi pubblica informazioni e di chi le richiede

- I nodi non sono in grado di rilevare il ruolo nella rete dei vicini con cui scambiano i messaggi

  Esempio: il vicino è il nodo che ha pubblicato l'informazione oppure un nodo che inoltra l'informazione per conto di un altro nodo?

- Tecniche utilizzate: modifica dell'identificatore del nodo sorgente di un messaggio.

  N riceve un messaggio che indica che il nodo P possiede un certo file F.

  N si sostituisce a P, come sorgente del file. N memorizza comunque l'identità del vero possessore di F.

  Le successive richieste ricevute da N per F, vengono propagate a P.

Freenet: Inserimento di Nuovi Peer

Quando un nuovo peer P si inserisce su una rete Freenet esistente:

- individua un peer PB sulla rete mediante una procedura di bootstrap
- inserisce PB nella propria tabella di routing, associandogli una chiave
- la chiave associata a PB può essere generata in modo casuale, il routing è in grado successivamente di adattare la tabella di routing di P, migliorando l'attendibilità dell'informazione contenuta
- mediante l'inoltro di queries sulla rete, P scopre nuovi vicini sulla rete. La tabella di routing di P viene automaticamente popolata, in modo dinamico, durante la permanenza di P sulla rete

Freenet: Inserimento di Nuovi Peers

Il nuovo peer P che si unisce ad una rete Freenet esistente deve

- "farsi conoscere" ad altri nodi della rete. In questo modo gli altri nodi possono inviare a P queries e nuovi file da memorizzare
- per "farsi conoscere", P invia al peer di bootstrap PB un messaggio di announcement, contenente la propria identità ed un opportuno valore del HTL
- PB sceglie, in modo casuale, un vicino a cui propagare l'announcement
- la propagazione continua, per HTL passi.

…. continua pagina successiva

Freenet: Inserimento di Nuovi Peer

- i nodi che ricevono l'announcement devono decidere di comune accordo quali chiavi assegnare al novo nodo
- le chiavi da assegnare a P possono essere definite in modo casuale, ma tutti i nodi devono concordare sui valori scelti
- i nodi eseguono un algoritmo distribuito che consente di concordare sul valore scelto per la chiave
- Cryptographic protocol: consente di definire un valore random in modo distribuito e di garantire che nessun nodo "malizioso" menta sul valore deciso collettivamente.

Freenet: Auto Organizzazione

Auto organizzazione: la rete si auto-adatta durante la vita del sistema, migliorando così la qualità del routing, in seguito a:

- Clusterizzazione delle informazioni contenute nelle tabelle di routing: ogni nodo "si specializza" naturalmente nella localizzazione di chiavi simili

  Esempio:

  il peer P è associato nelle tabelle di routing dei vicini alla chiave K.

  P riceve queries per chiavi simili alla K. Di conseguenza P diviene sempre più esperto nell'individuare chiavi simili a K.

- Replicazione trasparente dei dati richiesti di frequente.

  Esempio:

  un file memorizzato su un peer a Sydney è richiesto a Pisa. Il peer di Pisa lo memorizza nella propria cache. Eventualmente il file può essere replicato sui peers incontrati nel cammino da Sydney a Pisa

Freenet: Auto Organizzazione

- Clusterizzazione dei dati caratterizzati da chiavi simili nelle memorie dei peer:

  Esempio:

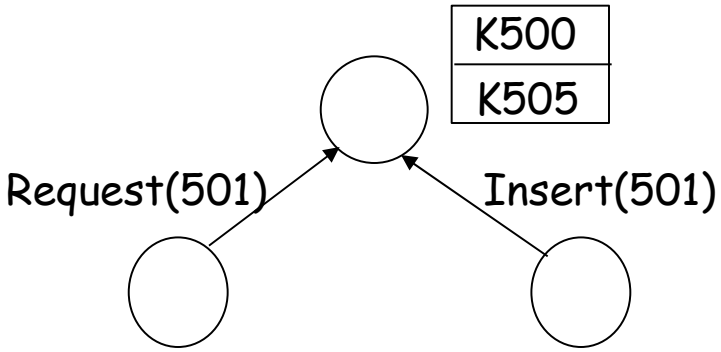
  il peer P memorizza la chiave 500

  il peer PN vicino a P riceve il messaggio *insert(505, htl),*

  PN probabilmente invierà a P il messaggio

- Quando i cammini seguiti dalle queries e dalle richieste di memorizzazione di "si incrociano" sulla rete i cluster tendono ad aumentare di dimensione

  Esempio:



Freenet: Gestione della Memoria

Capacità limitata di memoria del sistema:

- la memoria viene gestita dai peer secondo un approccio LRU (Least Recently Used)
- i files utilizzati meno di frequente vengono eliminati per primi per far spazio a nuove memorizzazioni
- le entrate delle tabelle di routing vengono lasciate inalterate, in modo da poter recuperare in seguito i files cancellati
- se tutti i nodi decidono di eliminare un file il file non risulta più disponibile nel sistema
- tutti i files sono criptati, ogni peer non conosce il contenuto della propria memoria condivisa