E-privacy 2003

riservatezza e diritti individuali in Rete

Firenze, 14 giugno 2003

Laboratorio Freenet

Marco A. Calamari - marco@freenetproject.org

Il Progetto Freenet

Firenze Linux User Group

Il Progetto Winston Smith

Indice

Cosa e’ Freenet ?

Cosa e’ Freenet

“Freenet e’ una rete adattativa di nodi peer-to-peer che si interrogano reciprocamente per immagazzinare e recuperare file di dati identificati da nomi (chiavi) indipendenti dalla locazione.”

Freenet e’ formata da server (nodi) paritetici; i nodi includono un proxy che permette di accedere al server Freenet con un form, utilizzando il protocollo HTTP.

“Freenet :
A Distribuited Anonymous Information Storage and Retrieval System”
I.Clarke et al.

… e tradotto in italiano ?

“Freenet e’ un sistema per scrivere e leggere file da Internet senza che si possa risalire a chi li ha scritti, chi li conserva sul disco e chi li recupera.”

Questo scopo viene raggiunto utilizzando il client (nodo) Freenet, che spezzetta, crittografa, duplica, disperde i contenuti del file, e riesce ad eseguire l’operazione inversa per recuperarli.

Freenet non permette di cancellare niente e non conserva informazioni su dove un file si trova.

Cosa trovo su Freenet ?

Freenet non e’ un applicazione ma un protocollo.

La cosa di gran lunga piu’ utile che troviamo su Freenet sono i freesites.

Sono gruppi di chiavi Freenet che, accedute via browser, si comportano quasi esattamente come un normale sito web.

Esiste un “indice” non ufficiale od esaustivo ma di grande reputazione ed utilita’, che elenca i freesite - “The Freedom Engine”

Cosa trovo su Freenet ?

Esistono due tipi di freesites:

DBR (Date Based Redirect), che visualizzano il contenuto riferito alla data odierna

Edition-based, che non cambiano ad intervalli fissi, ma visualizzano un avvertimento quando ne viene prodotta una edizione piu’ recente

Sono sicuri i Freesite ?

La navigazione su un freesite, normalmente anonima, puo' essere tracciata se un freesite trappola utilizza accorgimenti per tracciare il navigatore.

Freenet include filtri che rilevano gli accorgimenti noti ed avvertono il navigatore.

L'utilizzo di Freenet da parte di chi desidera anonimato non e' foolproof, ma deve essere fatto con attenzione.

"Per creare l'anonimato e' necessaria una complessa attivita' tecnica, per distruggerlo basta un click disattento."

Obbiettivi da raggiungere

- **Anonimato** sia per il produttore che per il fruitore dell'informazione
- Il sistema non deve avere elementi di controllo centralizzati o di amministrazione
- Il sistema deve essere robusto rispetto ai problemi hardware/software
- Il sistema deve "adattarsi" e mutare nel tempo
- Le performance devono essere paragonabili ad altri sistemi (WWW)

Modelli Peer-to-peer

- Modello centralizzato
  - Esempio : Napster
  - indice mantenuto da un autorità centrale - conoscenza globale dei dati (single point of failure)
  - contatto diretto tra richiedente e fornitore
- Modello decentralizzato
  - Esempio : Freenet, Gnutella
  - nessun indice globale – conoscenza locale dei dati (approximate answers)
  - contatti mantenuti da una “catena” di intermediari

Caratteristiche

- **Versione 0.5.2**
- **Realizzata in linguaggio java** - funzionante su differenti architetture
- Possiede una **interfaccia utente nativa** (inclusa in Fproxy) che permette di operare in maniera intuitiva, ma anche di controllare aspetti molto tecnici del nodo.
- **Datastore nativo crittografato** - non e' possibile cercare una categoria di contenuti, ma solo identificare un file dato

Caratteristiche

- **Routing adattativo** - il grafo delle connessioni logiche tra i nodi evolve nel tempo verso una stabilita' ed efficienza maggiore, ed i nodi stessi si specializzano .
- **Comportamento non deterministico** - il funzionamento di Freenet non e' completamente deterministico, e non consente di provare con certezza che un certo file presente nel datastore sia stato richiesto dal nodo locale e non da un altro nodo della rete

Caratteristiche

- **Resilienza della rete** - Freenet puo' perdere una rilevante percentuale di nodi senza un'apprezzabile riduzione di prestazioni, e la maggioranza dei suoi nodi senza cessare di funzionare
- **Comportamento "ecologico"** - l'informazione puo' essere inserita in Freenet ma non rimossa; puo' solo essere lasciata "morire" di morte naturale. L'informazione che viene richiesta si moltiplica su piu' nodi e si "avvicina" ai nodi che la richiedono; quella non richiesta scompare.

Caratteristiche

- **Anonimita’** sia di chi memorizza informazioni che di chi le recupera - nel caso si prevedano attacchi con memorizzazione del traffico sono necessarie cautele aggiuntive (Fproxy attraverso un tunnel SSL).

- **Autenticazione crittografica tra i nodi** - non e’ possibile “impersonare” un nodo gia’ noto alla rete sostituendosi ad esso

FEC splitfile

Meccanismo di suddivisione ridondante dei file per l'inserimento ed il recupero di file di grosse dimensioni (FEC splitfile di Onion Networks).

L'inserimento di un grosso file in Freenet e' problematico; con la suddivisione di un file in parti piu' piccole si risolve il problema dell'inserimento, ma basta l'impossibilita' di recuperare un pezzo ed il file e' perso.

L'algoritmo FEC (Forward Error Correction) moltiplica di un certo fattore (tipicamente 1,5) il numero di parti, ma aumenta la possibilita' di recuperare integralmente il file perche' non e' necessario recuperarne tutte le parti.

FEC splitfile - esempio

Supponiamo di avere un file di 10 Mb e di suddividerlo in 10 parti; supponiamo che la probabilita' di recuperare una chiave qualsiasi da freenet sia del 90%.

La probabilita di recuperare tutte e 10 le chiavi e' 0.90^10 = 0,3486 cioe' meno del 35%.

Se inserisco invece 15 parti ridondate con l'algoritmo FEC, la probabilita' di recuperare l'intero file (cioe' almeno 10 blocchi su 15) e' del 99.8%.

Quest'ultimo calcolo statistico e' lasciato all'abilita' matematica del lettore, oppure e' disponibile dietro modico sovrapprezzo 8) .

Cosa Freenet non puo' fare

- Non esiste attualmente la possibilita' di indicizzare le chiavi in modo da operare una ricerca intelligente.
- Il problema non e' risolto a livello di protocolli
- Esiste una proposta per la creazione e gestione di indici interni a Freenet (FASD, Kronfeld et al.) che pur descritta completamente a livello teorico, non e' stata ancora implementata

Cosa Freenet non puo' fare

- Non esiste un meccanismo "sicuro" di boot di un nuovo nodo senza possedere un minimo di informazioni sulla rete.
- Attualmente, per bootstrappare un nuovo nodo, bisogna conoscere l'indirizzo di almeno un nodo "affidabile" di Freenet.
- Si utilizzano un server web del Progetto e/o un file aggiornato di nodi distribuito insieme ai file di supporto di Freenet

Come funziona Freenet

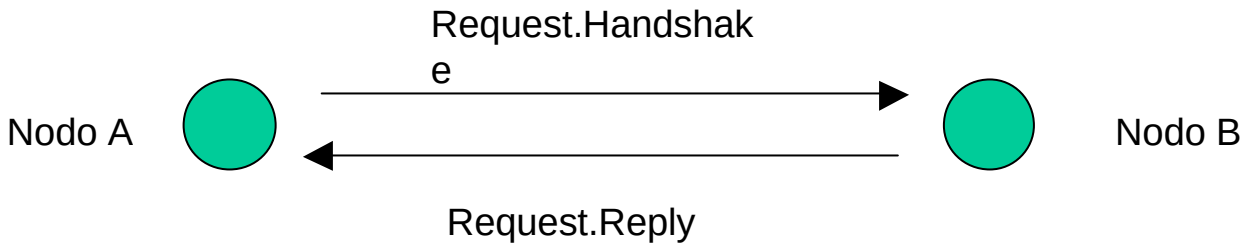
Come funziona Freenet

- **I nodi comunicano tra loro con un semplice protocollo connection-oriented chiamato FNP (Freenet Network Protocol), normalmente realizzato sopra il tcp/ip**

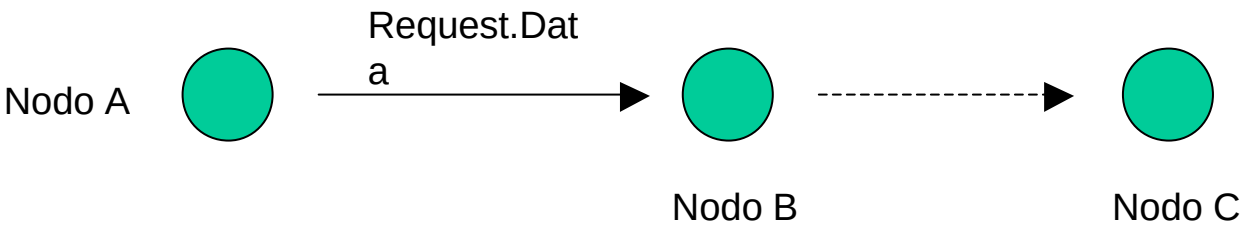
- **I client applicativi (e.g. Frost) che vogliono utilizzare i servizi Freenet di un nodo locale utilizzano un altro protocollo chiamato FCP (Freenet Client Protocol)**

Come funziona Freenet

Fase di Handshake

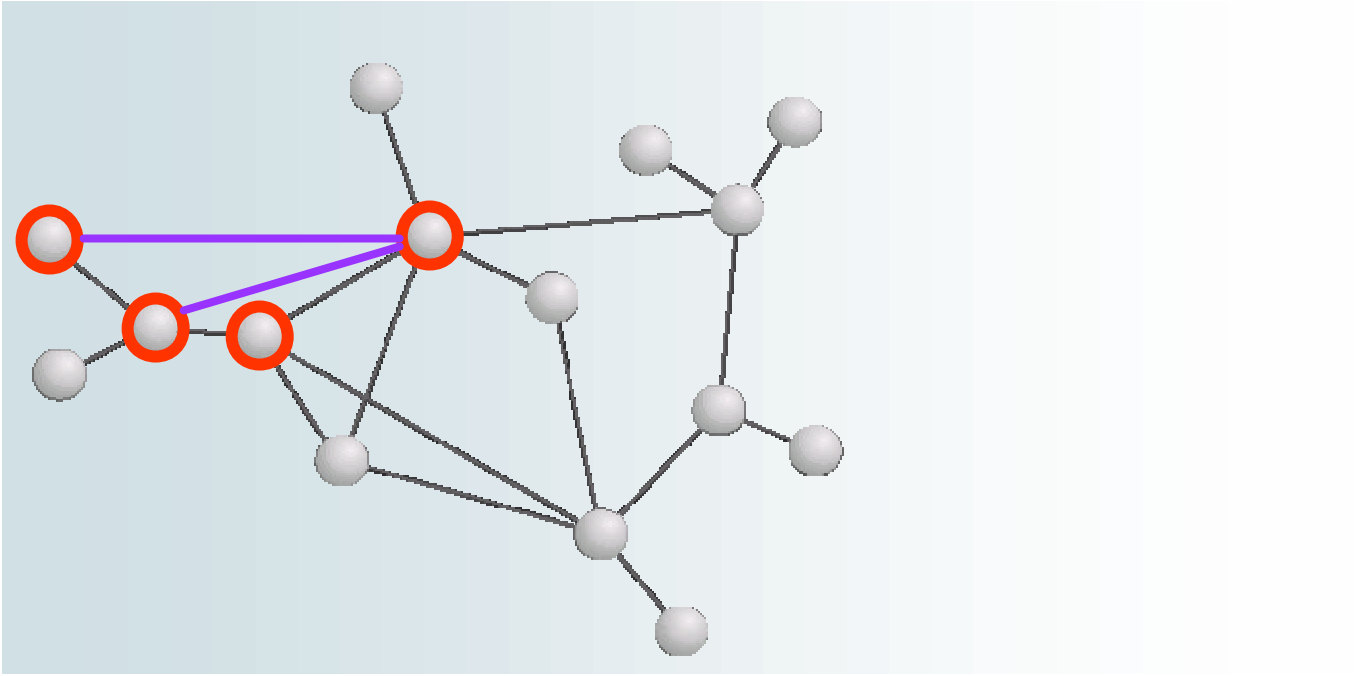


Fase di richiesta dati



Come funziona Freenet

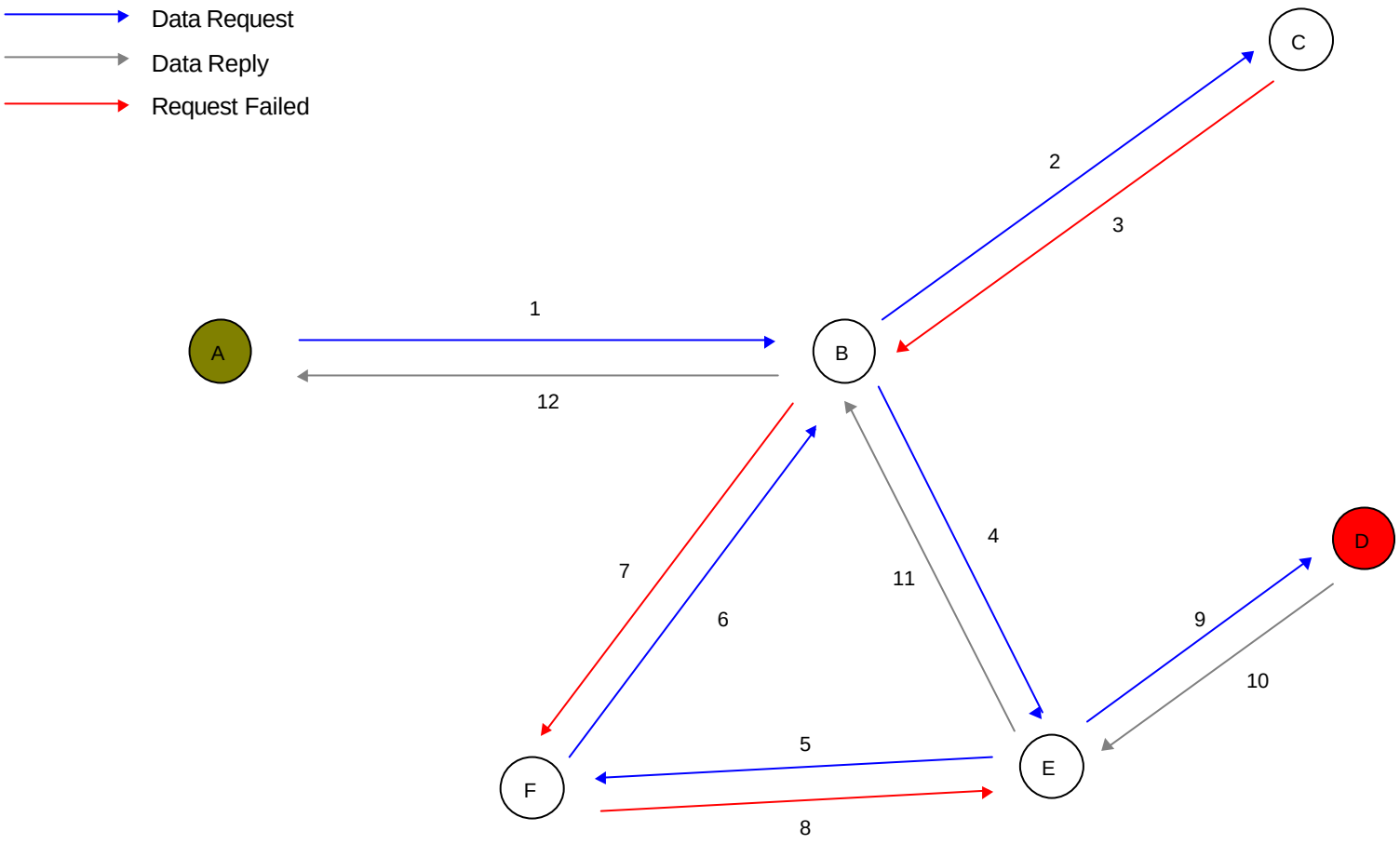
- I nodi comunicano tra loro sulla base di una conoscenza locale dinamica dei nodi limitrofi
- Ogni nodo richiede una chiave, nell'ordine, ai nodi limitrofi



Come funziona Freenet

- Un nodo che riceve da un confinante la richiesta di una chiave che ha precedentemente cercato e non trovato la rigetta immediatamente.
- Un nodo che deve inserire una chiave, prima la ricerca per evitare una collisione, e successivamente la inserisce
- La “profondita’” della ricerca o dell’inserimento di una chiave e’ data dall’HTL (hops to live)
- Ogni nodo che deve passare una richiesta decrementa l’HTL di 1

Come funziona Freenet



Come funziona Freenet

- Ogni nodo memorizza le chiavi “alla rinfusa” in un database che viene denominato “datastore
- Una chiave esiste solitamente in piu’ copie su piu’ nodi, in dipendenza dalla profondita’ di inserimento della richiesta originale
- Ogni nodo che, dopo aver trasmesso una richiesta che ha avuto successo, riceve la chiave da ripassare al nodo richiedente, ne fa una copia nel datastore locale

Come funziona Freenet

- Un “rumore di fondo probabilistico” viene inserito in tutte le decisioni di routing (variazione dell’HTL, possesso della chiave, etc.) per impedire che un eventuale registrazione del traffico possa far risalire al nodo che ha effettuato la richiesta o l’inserimento originali, e permettere all’operatore del nodo la ripudiabilita’ di un’eventuale attribuzione di responsabilita’ del contenuto del datastore.

Come funziona Freenet

- I singoli datastore vengono gestiti con un watermark, sulla base della data e del numero degli accessi alle singole chiavi
- Le chiavi “popolari” si moltiplicano e si spostano “vicino” ai nodi che le richiedono
- Le chiavi “impopolari” scompaiono
- Si tratta di un comportamento “ecologico” che permette di realizzare un sistema in cui non esiste il comando “delete”

Come funziona Freenet

- I singoli nodi si “specializzano” nel memorizzare alcune chiavi, basandosi su una “distanza lessicale” che viane calcolata utilizzando un hash del contenuto della chiave, e specializzandosi in un segmento di essa
- Le decisioni di routing delle richieste possono essere fatte in maniera intelligente, poiche’ i nodi pubblicizzano il segmento di spazio delle chiavi in cui sono “specializzati”

Meccanismi crittografici

Le chiavi di Freenet

- I file in Freenet sono associati e memorizzati utilizzando oggetti detti “chiavi” :

  KSK (keyword signed key)

  CHK (content hash key)

  SSK (signed subspace key)

- Nota : la funzione hash utilizzata è lo SHA-1 a 160 bit mentre l’algoritmo asimmetrico di cifratura è il DSA

La chiave KSK

- E’ la chiave più semplice e user-friendly
  - Esempio -> freenet:KSK@mio_file.txt
  - La stringa descrittiva (mio_file) viene utilizzata per generare una coppia di chiavi pubblica/privata (algoritmo DSA)
  - La chiave pubblica viene utilizzata per produrre l’hash associato al file inserito (SHA-1)
  - La chiave privata viene utilizzata per “firmare” il file inserito.

La chiave CHK

- E’ derivata dall’hash del contenuto del file corrispondente. Tutti i file sono chiavi CHK
- Il file viene inoltre criptato utilizzando una chiave generata in modalità random
- Vengono pubblicati sia l’hash che la chiave di decrittazione
  - Esempio -> freenet:CHK@foto.gif
  - Una volta inserito, il dato potrà essere richiesto fornendo la seguente stringa :

CHK@zdfaGT….,fpR12…..

La chiave SSK

- Costruzione di un “namespace” personale
  - Creiamo una coppia di chiavi pubblica/privata di tipo SSK
  - Utilizzeremo la chiave privata per inserire documenti “sotto” il nostro namespace
  - Pubblicheremo la nostra chiave pubblica per rendere accessibili i file pubblicati
  - Esempio -> SSK@public_key/musica/song1.mp3

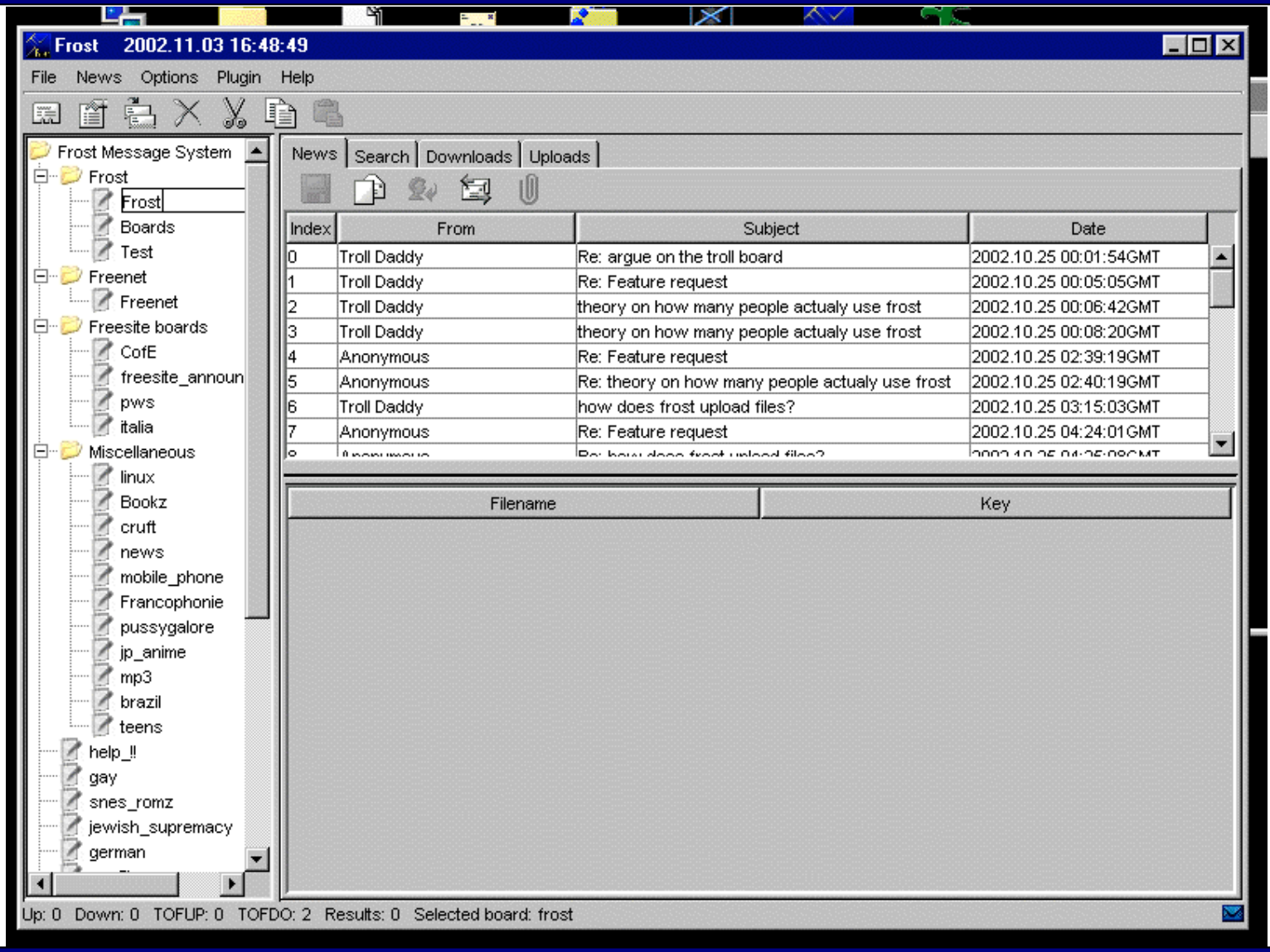
    SSK@public_key/musica/song2.mp3

Client ed applicazioni

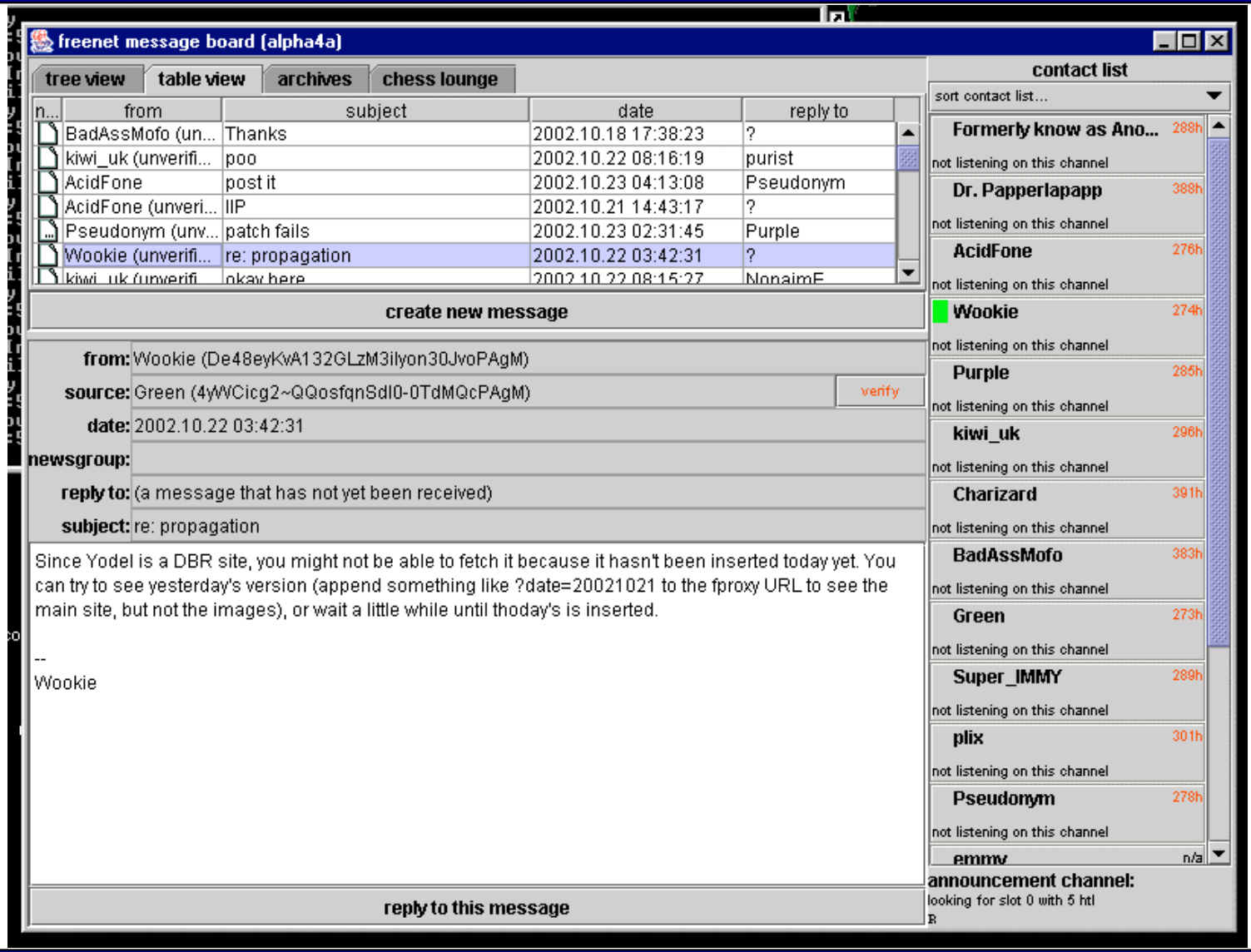
Client ed applicazioni

- **Frost** - client grafico per la messaggistica, il chat e la condivisione sicura di files
- **FMB Freenet Message Board** - messaggistica sofisticata, scambio di file e scacchi anonimi
- **Espra** - creazione e gestione di cataloghi
- **Freeweb** - client grafico per la creazione di freesite
- **Manifest** - client a linea comandi per la gestione di chiavi e freesite
- **FCPtools** - client a linea comandi per la gestione di chiavi e freesite

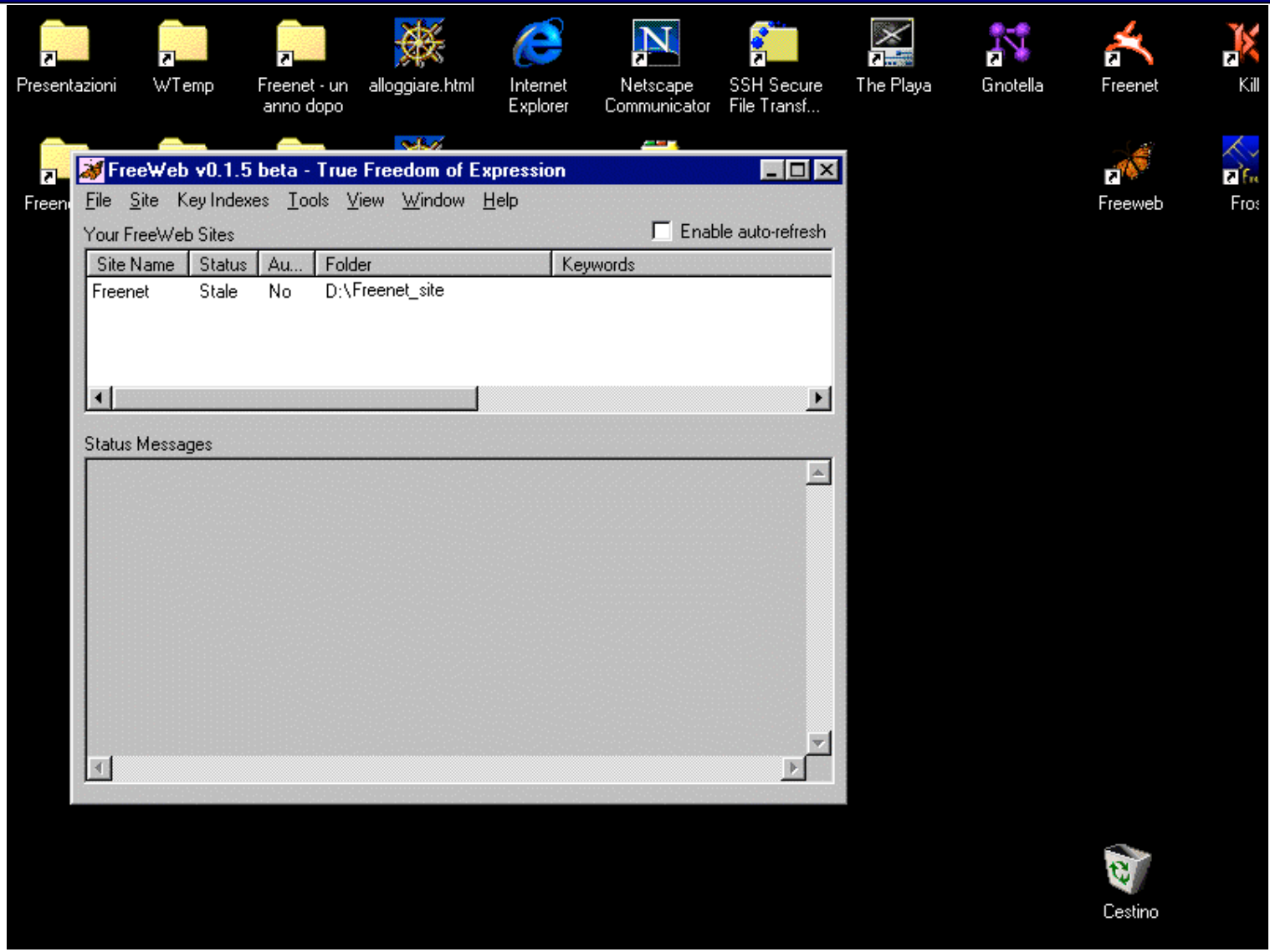
Frost



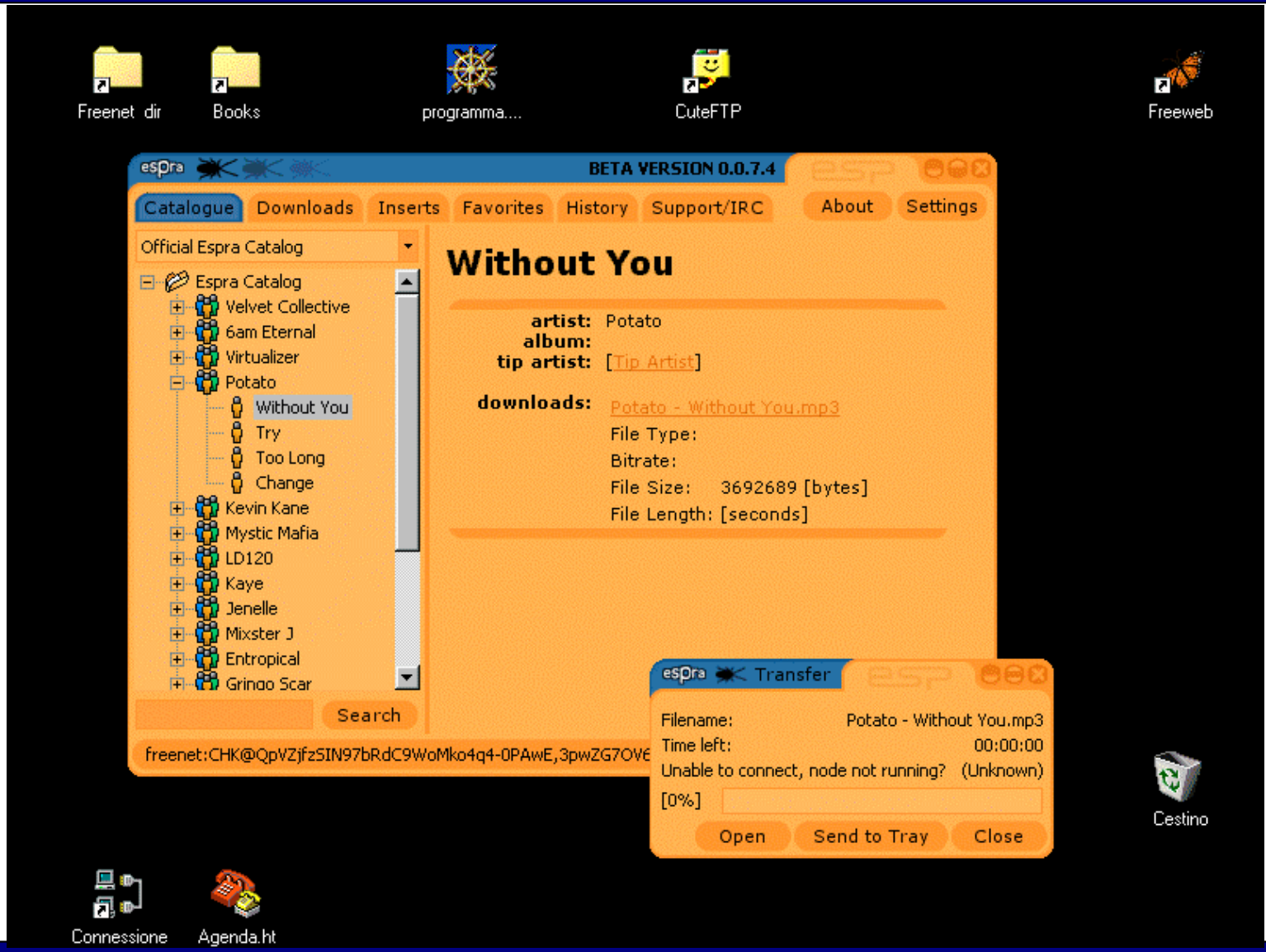
FMB



FreeWeb



Espra



Obbiettivi futuri

pagina

pagina

Obbiettivi futuri

- Formazione di un **gruppo di sviluppo piu' grande** e piu' strutturato, che applichi metodi di sviluppo piu' formalizzati
- **Documentazione esaustiva di protocolli**, API e metodi di routing ("Ma perche' ? Non lo fanno gia' ?" "No!")
- Studio sistematico delle **metodologie di attacco** alla rete Freenet
- **Diffusione dell'utilizzo** di Freenet e sviluppo di nuovi client che la utilizzino come mezzo di trasporto e/o memorizzazione.

N.d.A. - malgrado Freenet 0.5 sia incomparabilmente migliorata in questo ultimo anno, gli obbiettivi che elencavo l'anno scorso sono rimasti gli stessi.

Bibliografia

pagina

pagina

Bibliografia

- "**Freenet : A Distribuited Anonymous Information Storage and Retrieval System**" - I.Clarke et al.
- "**Performance in Decentralized Filesharing Networks**" - T. Hong
- "**Advanced Routing on Freenet**": (Serapis) - Shu Yan Chan
- "**FASD: A Fault-tolerant, Adaptive, Scalable, Distributed Search Engine**" - Amr Z. Kronfol, Princeton University May 6, 2002

I documenti sono reperibili sul sito del progetto http://freenetproject.org

Grazie a tutti per l’attenzione

per maggiori informazioni: **marco@freenetproject.org**

mail list su Freenet in italiano

http://lists.firenze.linux.it/mailman/listinfo/freenet-list

Sito ufficiale Freenet in italiano

http://www.freenetproject.org/

Il progetto Winston Smith

freenet:SSK@Dgg5lJQu-WO905TrlZ0LjQHXDdIPAgM/pws/9//