Lezione n.14

IL PROTOCOLLO EDONKEY - eMule

Materiale didattico distribuito a lezione

Laura Ricci

eMule: UN PO' DI STORIA....

- Origine:il protocollo eDonkey, ilclient edonkey2000 (ed2k)
- E-donkey: rete peer-to-peer per la condivisione di file di qualsiasi tipo
- I client eDonkey si connettono alla rete per condividere file.
- I server eDonkey hanno una funzione simile al server NAPSTER:permettono agli utenti di localizzare i file all'interno della rete.
- Client e server eDonkey disponibili per i sistemi Windows, Macintosh, Linux..
- Aggiunta dinamica di servers alla rete.
- A causa del costante cambiamento della rete dei server, i client devono aggiornarne costantemente la lista.

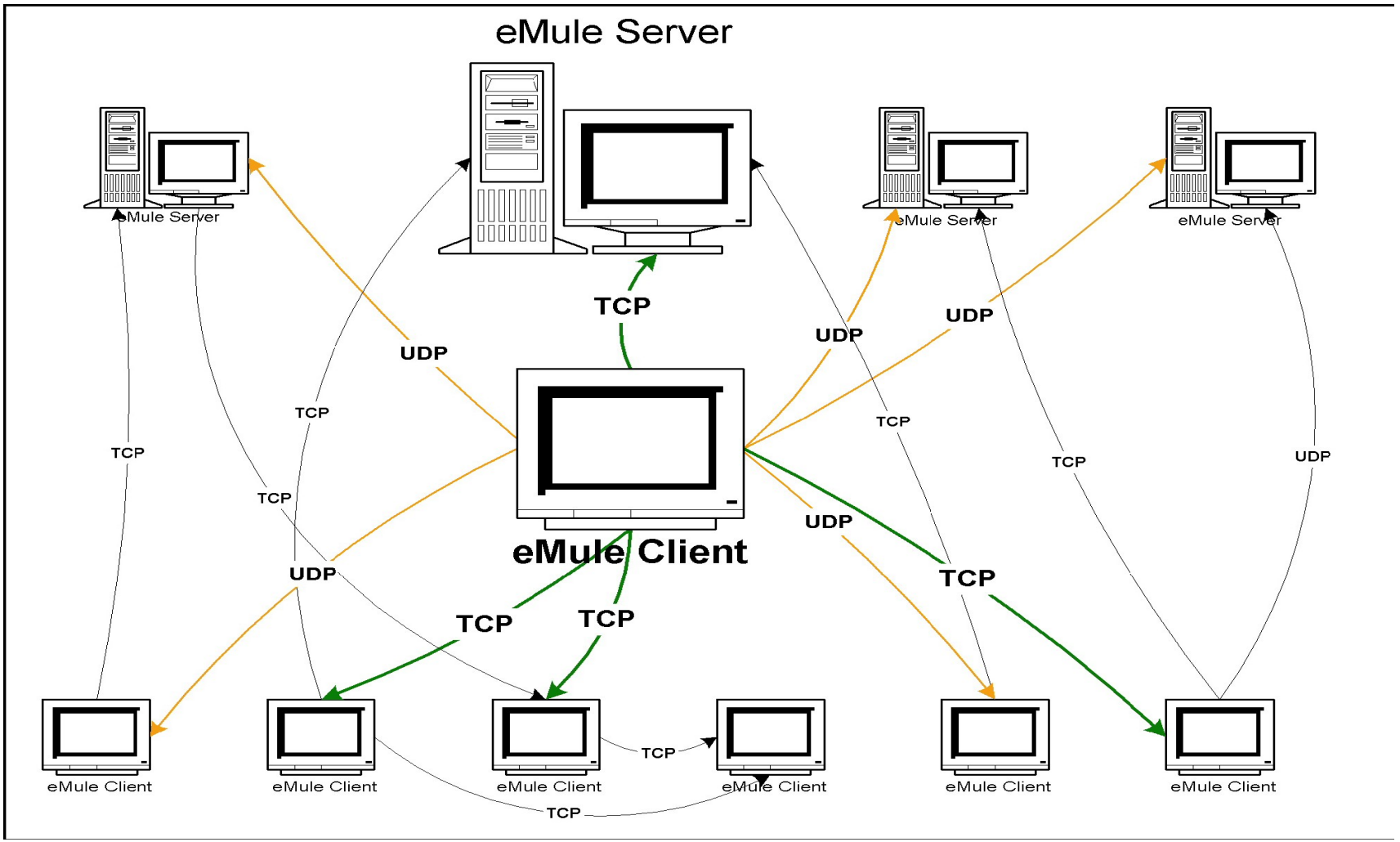
eMule: UN PO' DI STORIA.....

- eMule: il nome del progetto sottolinea sia le somiglianze che le diversità con il programma eDonkey2000:
- eMule (mulo) simile aad un donkey (asino). Emule "emula" le funzionalità di eDonkey.
- sviluppato nel maggio 2002 dal programmatore tedesco Merkur, come un programma eDonkey-compatibile ma con molte più funzioni.
- sfrutta due diversi tipi di overlay networks
  - La rete eDonkey (ed2k), basata sul modello client/server.
  - La rete KAD (presente a partire dalla versione 0.42)
    - implementa KADEMLIA, una DHT con caratteristiche simili a Pastry.
    - è una rete serverless in cui il carico di lavoro è distribuito a tutti i client connessi invece che ad un unico server
- Utilizzato ogni giorno in medi da 3 milioni di utenti che condividono circa 500 milioni difiles

eMule: LA RETE ED2K

- Basato su un sistema P2P orientata al file sharing
  - basata su una estensione del protocollo eDonkey
  - architettura Client/Server
  - Utilizza sia il protocollo TCP che il protocollo UDP
- Client open source: miglioramenti continue delle versioni del client eMule
- Il protocollo definisce le interazioni tra
  - Client e servers
  - Client e clients
- Strategie implementate nel client
  - Gestione delle code
  - Sistema dei crediti
  - Ordinamento delle parti di file scaricate
  - Gestione (recovery) di parti di file corrotte

LA RETE eMule ed2k



IDENTIFICAZIONE DELLE ENTITA' NELLA RETE eMule

USER ID (User Hash)

- stringa di 128 bits (16 bytes) ottenuta concatenando numeri random
- utilizzato per implementare il sistema dei crediti
- assegnato la prima volta che eMule viene lanciato, non viene modificato nelle sessioni successive
- memorizzato nel file preferences.dat nella directory emule/config, identifica il client/host

CLIENT ID

- Stringa di 32 bits (4 bytes)
- Valida solamente per la durata di una sessione
- Può essere assegnato un ID ALTO oppure un ID BASSO a seconda del tipo di connessione tra l'utente e la rete

FILE HASH

- Stringa hash di 128 bits calolata mediante l'uso di MD4
- Assegnata dal sistema non appena il file viene messo in condivisione
- Identifica univocamente il file all'interno della rete
- Può essere individuato nella finestra file condivisi, colonna FILE ID

IL PROTOCOLLO CLIENT SERVER

- Ogni client si connette al massimo ad un server per volta
- L'utente può scegliere il server da una lista oppure lasciare la scelta automaticamente al sistema. Successivamente, il server può essere cambiato solo su intervento dell'utente
- Si stabiliscono due connessioni TCP:una dal client al server e viceversa
- Nel momento in cui la connessione viene stabilita il server

  - identifica il client
  - decide se assegnare al client un ID BASSO oppure un ID ALTO, valido per l'intera sessione e determinato in base alla raggiungibilità del client dall'esterno
  - può decidere di rifiutare la connessione con il client (ad esempio quando il numero di connessioni gestite dal server supera una soglia predefinita)

ASSEGNAZIONE DEL CLIENT ID

- Il Server assegna ad un client C un ID BASSO se C non può accettare connessioni in ingresso, ad esempio si trova amonte di un NAT o di un firewall.
- ID ALTO il client C consente agli altri client dalla rete eMule di connettersi liberamente alla porta TCP eMule 4662
- ID ALTO: viene calcolato a partire dall'indirizzo IP dell'host
  - Se l'indirizzo IP del client è X.Y.Z.W

    ID = X+ 256*Y + 256*256*Z +256*256*256*W
- L'ID (alto o basso) viene visualizzato al momento della connessione col server

  06/05/2007 0.15.35: WARNING DonkeyServer No2 (62.241.53.16:4242) – Your 20603 port is not reachable. Please review your network config.

  06/05/2007 0.15.35: Connessione stabilita con: DonkeyServer No2

  06/05/2007 0.15.35: Caricamento fonti dal nuovo server in corso per tutti i file con meno di 400 fonti

  06/05/2007 0.15.35: Il nuovo ID è 4241605

  06/05/2007 0.15.35: Ricevuti 1 nuovi servers

ASSEGNAZIONE DEL CLIENT ID

- Il Server assegna ad un client C un ID basso se C non può accettare connessioni in ingresso (ad esempio si trova amonte di un NAT o di un firewall)
- ID ALTO il client C consente agli altri client dalla rete eMule di connettersi liberamente alla porta TCP eMule 4662
- ID ALTO: viene calcolato a partire dall'indirizzo IP dell'host
  - Se l'indirizzo IP del client è X.Y.Z.W

    ID = X+ 256*Y + 256*256*Z +256*256*256*W
- L'ID (alto o basso) viene visualizzato al momento della connessione col server

  06/05/2007 0.15.35: WARNING DonkeyServer No2 (62.241.53.16:4242) – Your 20603 port is not reachable. Please review your network config.

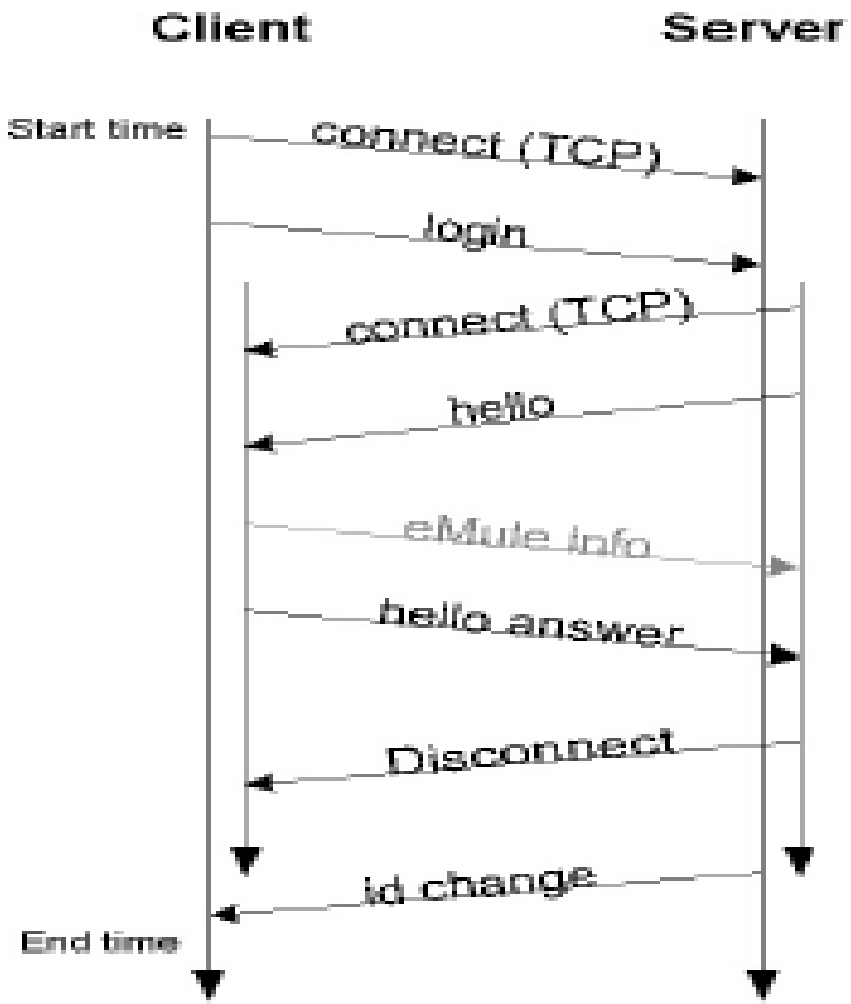
  06/05/2007 0.15.35: Connessione stabilita con: DonkeyServer No2

  06/05/2007 0.15.35: Caricamento fonti dal nuovo server in corso per tutti i file con meno di 400 fonti

  06/05/2007 0.15.35: Il nuovo ID è 4241605

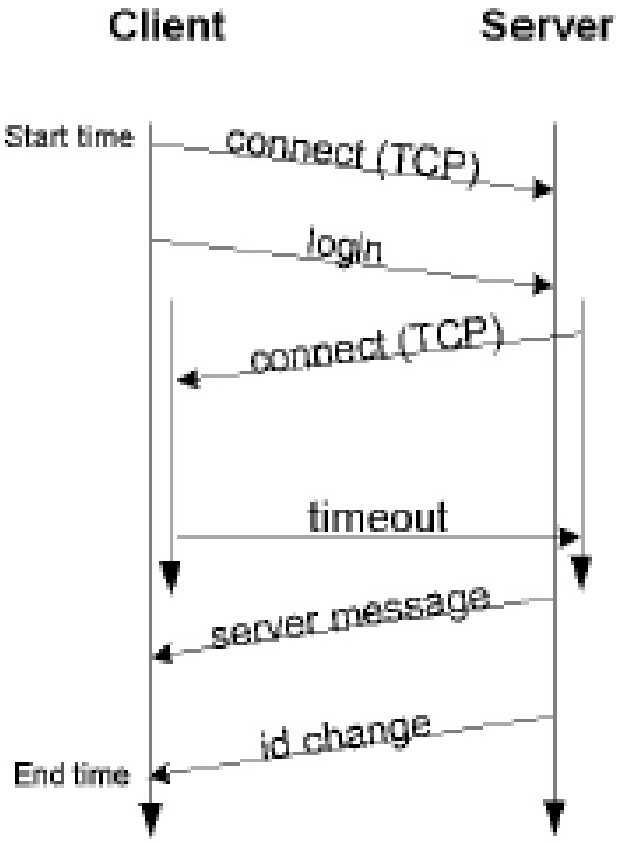
  06/05/2007 0.15.35: Ricevuti 1 nuovi servers

ASSEGNAZIONE CLIENT ID ALTO



ID-CHANGE = assegnazione dell'id al client

ASSEGNAZIONE CLIENT ID BASSO

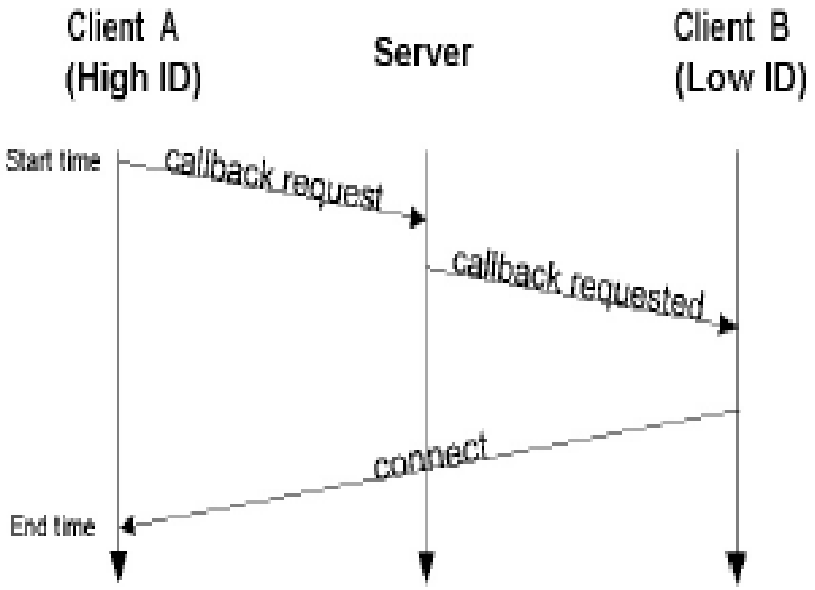


- Il server non riesce a connettersi al client
- Il server invia sulla connessione aperta dal client un messaggio del tipo:

Warning: You have a low ID. Please review your network configuration and/or your settings

- la fase di connessione si conclude con un messaggio di ID CHANGE che assegna al client un ID basso

ID BASSI: IL MECCANISMO DELLE CALLBACKS



- Consentono ad un utente A con ID alto A di connettersi ad un utente B di ID
- basso
- A chiede a B una 'callback'
- A e B devono essere connessi allo stesso server

ID BASSI: LIMITAZIONI

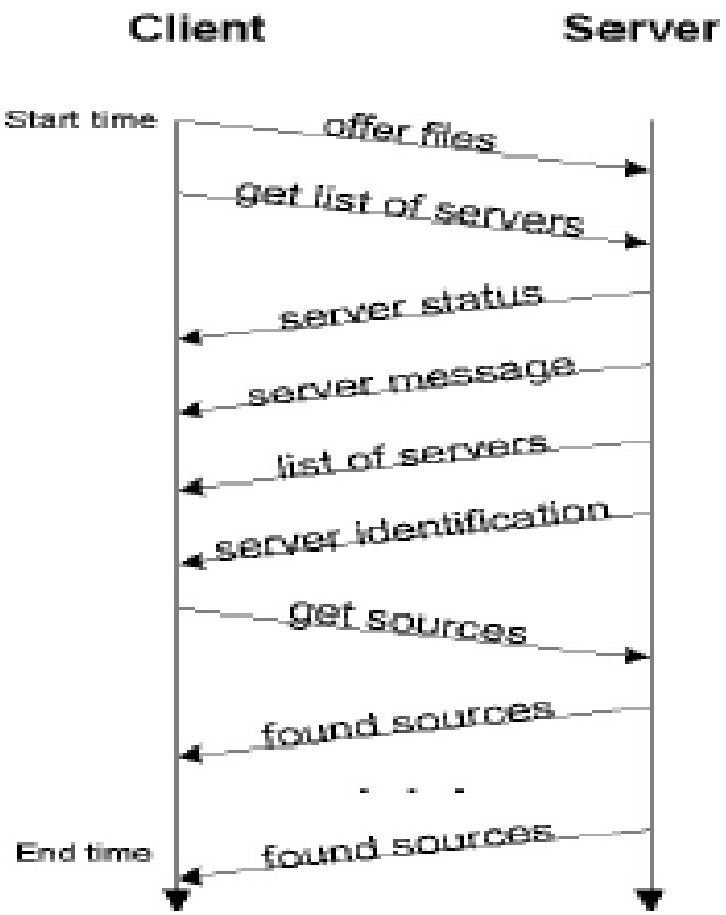
- I servers accettano un numero limitatodi connessioni con utenti caratterizzati da un ID basso, perchè una callback coinvolge ilserver stesso
- Due client caratterizzati da ID basso non possono stabilire un collegamento tra di loro (nelle prime versioni di e-mule il server esegiva funzionedi tunneliing, ora eliminate per non caricare troppo i servers)
- Un client con ID basso può condividere files solo con altri clients caratterizzati da un ID alto e collegati allo stesso server
- Le fonti di un client di ID basso non vengono diffuse agli altri servers

COME FARE PER OTTENERE UN ID ALTO

- Configuare opportunamente il firewall
- Aprire le porte sul router/NAT
  - Configuarare il router/NAT in modo che tutte lecomunicazioni provenienti dall'esterno e dirette alle porte TCP 4662 ed UDP 4672 siano instradate verso l'IP corrispondente al computer su cui è in esecuzione eMule
  - Se esistono più clients collegati al router mediante una rete locale, occorre associare alle diverse istanze di eMule porte diverse e "aprire" tutte queste porte sul router
  - E' inoltre necessario 'spegnere' il DHCP, perchè questo assegnerebbe un indirizzo IP diverso ogni volta che il sistema viene reinizializzato
  - L'apertura delle porte spesso può essere effettuata solo dall'amministratore della rete

CLIENT-SERVER START UP

MESSAGGI DI START UP



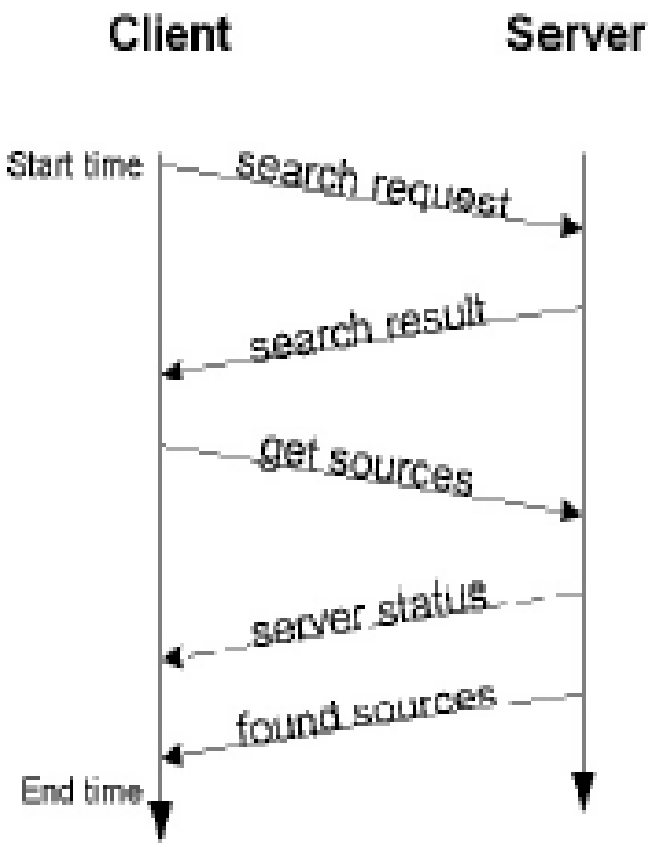
- Il client invia la lista dei files che intende condividre

- Il server invia una lista di servers da lui conosciuti

- La figura assume che il client C possieda una lista L di files richiesti al momento della connessione con il server.

- Il server invia automaticamente nuove fonti per tutti i files in L per cui C possiede un numero limitato di fonti (valore soglia prestabilito)

INTERAZIONE CLIENT-SERVER:RICERCA DI FILES



- Il client invia una query che può contenere operatori booleani AND, OR, NOT

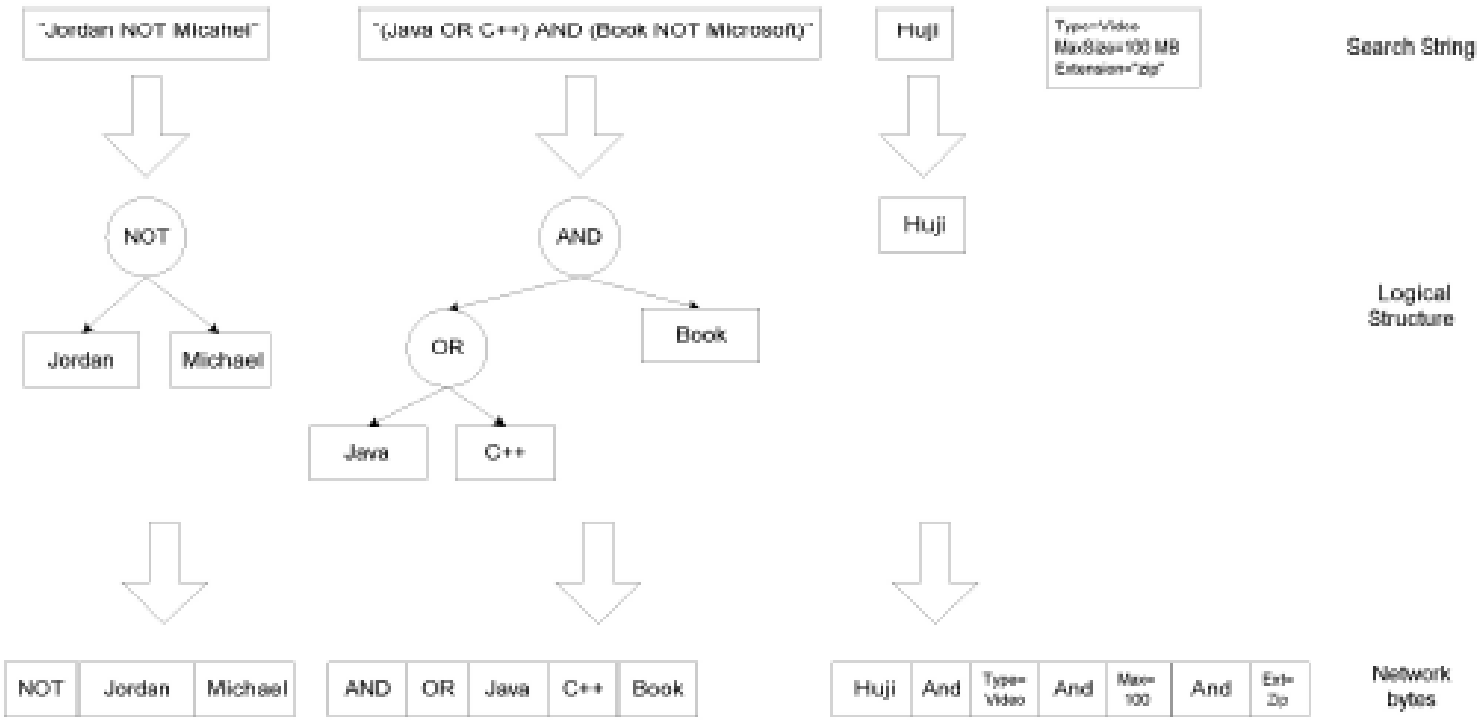
- il server risponde con una lista di files che soddisfano la query

- il client sceglie uno o più files da scaricare

- il server comunica una lista di fonti per i files prescelti (ogni fonte può possedere tutto il file o una parte di esso)

- status message contiene informazioni sul numero di utenti e di files gestiti dal server

INTERAZIONE CLIENT/SERVER: RICERCA DI FILES



- ogni query booleana viene rappresentata mediante un albero
- l'albero viene linearizzato mediante una visita anticipata

INTERAZIONI UDP TRA CLIENT E SERVERS

La comunicazione UDP tra client e server viene utilizzata per:

Keep alive:

- il client verifica periodicamente lo stato dei servers nella sua lista,
- La richiesta del client include un numero random su cui viene fatto l'eco da parte del server
- Ogni volta che il client invia un keep alive al server, incrementa un contatore *C*
- Ogni volta che il client riceve un messaggio di qualsiasi tipo dal server, *C* viene decrementato
- Se *C* raggiunge un valore soglia, il server viene considerato offline e viene escluso dalla lista dei servers

Ottimizzazione di ricerca delle fonti

- Se il numero di fonti per un certo files ricevute dal server di riferimento risulta sotto una certa soglia, il client invia pacchetti UDP agli altri servers nella sua server list per ottenere ulteriori fonti

Ottimizzazione ricerca dei files Configurabile mediante l'interfaccia del client

COMUNICAZIONE CLIENT/CLIENT

- La maggior parte dei messaggi viene spedita su connessioni TCP
- Viene stabilita una diversa connessione TCP per ogni coppia [file, client]
- Connection startup: prevede un handshake simmetrico in cui i due clients si identificano mediante una procedura di Secure User Identification
- Ogni client
  - mantiene una coda di upload
  - Coda di upload = coda a priorità
  - in ogni istante serve un numero limitato di download, in modo da non esaurire tutta la banda a sua disposizione
  - quando un client richiede il download di un file, la sua richiesta
    - viene servita immediatamente se la coda di upload è vuota ed il numero di download attivi non supera una certa soglia
    - altrimenti la richiesta viene inserita nella coda di upload

INTERAZIONE TRA CLIENTS: GESTIONE DELLE CODE

- Priorità = (rating * waiting time) / 100
- La priorità di un client nella coda di upload dipende da
  - il tempo di attesa in coda
  - un rating che dipende
    - dal credito acquisito (vedi lucidi successivi) dal client CD che effettua il download presso il client CU che effettua l'upload (credito acquisito da CD per precedenti upload effettuati da CD nei confronti di CU) (1-10)
    - dalla priorità del file che si sta scaricando (0.2-1.8)
    - In genere, rating = credito acquisito * priorità del file
- Il download inizia quando il client raggiunge la prima posizione della cosa
- Il downlaod continua finchè
  - Il client interrompe il downlaod
  - Preemption: un client con una priorità più alta richiede il downlaod

INTERAZIONE TRA CLIENTS: GESTIONE DELLE CODE

- Nel client eMule, nella finestra downloads è possibile individuare, accanto al nome del file in download il suo QR (Queue Rank)
- QR= 100, esistono 100 utenti in coda prima che devono essere serviti prima che la mia richiesta venga presa in considerazione
- QR da un'approssimazione del tempo di attesa, infatti è possibile che altri utenti mi superino sulla coda perchè possiedono un numero maggiore di crediti

INTERAZIONI TRA CLIENTS: IL SISTEMA DEI CREDITI

- Sistema di crediti: introdotto per incentivare la condivisione dei files da parte degli utenti
- Quando un client *C1* effettua l'upload di un file verso un client *C2*, *C1* viene 'ricompensato' da *C2* mediante l'attribuzione di un credito
- I crediti non sono globali,
- Un credito è un valore intero nell'intervallo 1-10 che viene assegnato ad ogni coppia ordinata (client, client)
- Un client memorizza in un proprio file i crediti degli altri clients da cui ha scaricato in precedenza qualche file
- I crediti di proprietà di un client *C* non vengono quindi memorizzati da *C*, ma dai clients che gli hanno concesso il credito, perchè hanno scaricato qualche file da *C*
- Non è quindi possibile visualizzare i propri crediti
- Viene introdotto un meccanismo di identificazione sicura dei clients per prevenire la richiesta illecita di crediti da parte di clients che non li posseggono

INTERAZIONE TRA CLIENTS: IL SISTEMA DEI CREDITI

- eMule per ogni utente tiene traccia, per ogni utente, di tutti i Megabyte forniti, per un periodo di 5 mesi .
- crediti mantenuti in un file nella cartella di configurazione di e-muale
- All'avvio eMule cancella i crediti scaduti

CALCOLO DEL VALORE DEI CREDITI

- Il valore dei crediti accumulati da un client *C1* presso un cliente *C2* è calcolato come il minimo tra i seguenti due valori

  - TotaleUploads * 2 / Totale Downloads
  - SQRT (Totale Uploads +2)

dove

- Totale Uploads è il numero totale di Mbytes inviati da *C1* a *C2*
- Totale Downloads è il numero totale di Mbytes inviati da *C2* a *C1*

ovvero posso riscuotere dei crediti presso un client se gli ho fornito più files di quanti ne abbia scaricati

- Condizioni particolari
  - Se totale Upload < 1MB il valore del credito è 1
  - Se il totale Download = 0, il valore del credito è 10

CALCOLO DEL VALORE DEI CREDITI

Esempio: analizziamo il punteggio che il mio client eMule attribuisce ad un altro client C.

Se ho ricevuto in passato 10MB da C e ne ho inviati 1:

FORMULA 1

- crediti = (Megabytes ricevuti * 2) / bytes inviati

  Crediti di C = (10 * 2) / 1 = 20 (cioè 10 dato che il massimo è 10)

FORMULA2

- crediti = SQRT (Megabyte ricevuti + 2)
- Crediti di C = SQRT (10 + 2) = circa 3.5

infine eMule sceglie il valore più basso fra i due calcolati quindi 3.5.

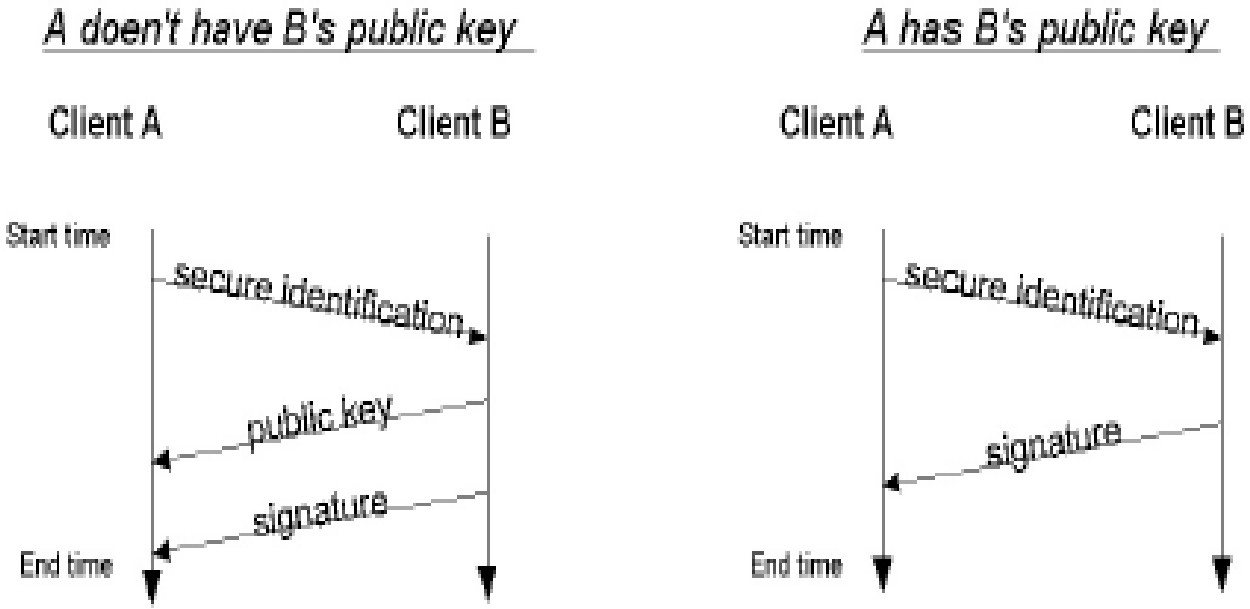
SECURE CLIENT IDENTIFICATION

- E' necessario utilizzare un sistema di identificazione sicura dei clients per evitare che un client si impossessi dei crediti destinati ad un altro client

- Basato su un sistema a crittografia asimmetrica (RSA)

- Supponiamo che un client B si deva autenticare presso il client A
  - B invia la sua identità e la firma generando la firmamediante la sua chiave privata,
  - A reperisce la chiave pubblica di B
  - A utilizza la chiave pubblica di B per decodificare la firma di B e verificare l'identità di B
  - eMule crea in automatico alla prima connessione la coppia chiave pubblica-chiave privata, che viene memorizzata nella cartella config di eMule.

SECURE CLIENT IDENTIFICATION



INTERAZIONE TRA CLIENTS: SCAMBIO FONTI

- Una connessione diversa per ogni file
- A invia una query relativa al file che intende scaricare
- B risponde con informazioni relative al file richiesto

INTERAZIONE TRA CLIENTS: SCAMBIO DELLE FONTI

- Ogni volta che richiedo il download di un file, eMule ricerca le fonti per quel file tramite i servers contenuti nella lista del client
- La ricerca avviene subito sul server di riferimento e, ad intervalli più lunghi, sugli altri servers della lista (mediante comunicazione UDP)
- Inoltre, quando si contatta una fonte per il download, il client richiede alla fonte contattata la lista delle fonti in suo possesso per il medesimo file
- Ciò avviene sia se la fonte contattata completa sia se solo alcune parti di quel file sono disponibili su quella fonte..
- Gli stessi client forniscono fonti per i files condivisi

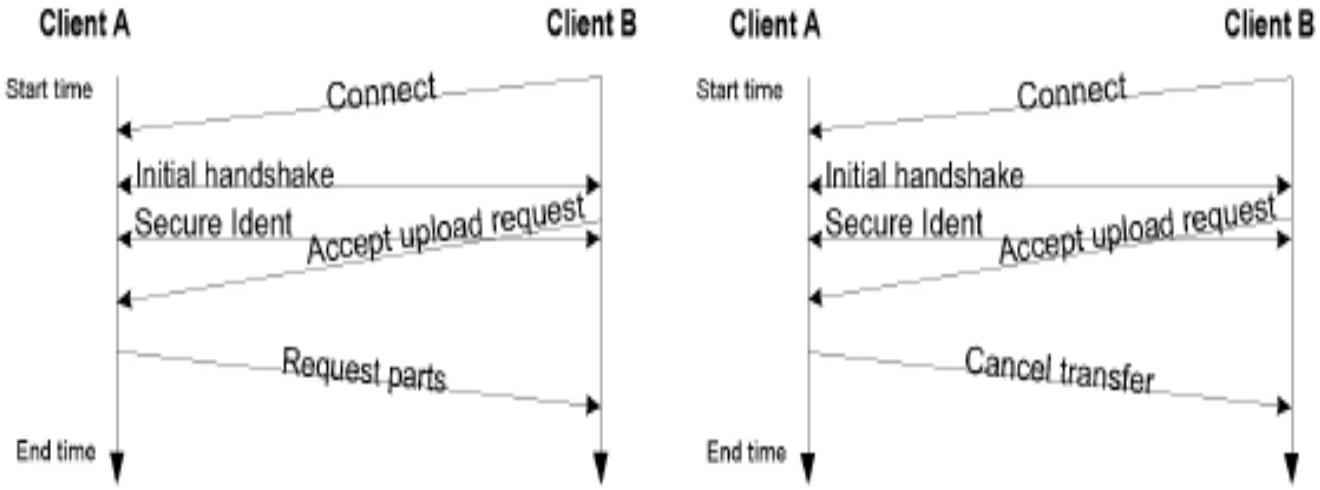
VANTAGGI

- Ricerca alternativa di fonti: percentuale di fonti contattate su quelle disponibili nella rete eMule molto prossima al 100%.
- Maggiore velocità di contatto, non dovendo passare per i server.
- Minor carico di lavoro dei server, con evidente miglioramento delle prestazioni del sistema.

TRASFERIMENTO DEI DATI TRA I CLIENTS

- Ogni file presente nella rete eMule viene decomposto in parti e quindi ogni parte viene frammentata
- Un client può scaricare fragmenti diversi dello stesso file da utenti diversi
- Quando inizio il download di un file, vengono contattate più fonti e la richiesta viene inserita nelle code di upload di tutte le fonti
- Quando il client *C* ha completato il download dell'intero file, non lo notifica agli altri clients
- Quando un client invia il messaggio di upload a *C*, la proposta di uplaod viene rifiutata da *C*

TRASFERIMENTO DI FILES TRA CLIENTS



- Quando A raggiunge la prima posizione nella coda di upload di B

B apre una connessione con A

- B chiede una connessione ad A

- A può accettare la connessione , oppure rifiutarla, se il file è già stato scaricato da altre fonti

TRASFERIMENTO DI FILES TRA CLIENTS

- Ogni file condiviso dalla rete eMule viene scomposto
  - in parti da 9.28 MB
  - Ogni parte è a sua volta divisa in chunks da 180K bytes
- Il client che effettua il download determina quali parti scaricare del file selezionato
- Le richieste di download si riferiscono a parti di files
- Il client determina l'ordina con cui le parti vengono scaricate e da quale client vengono scaricate
- Tutti i blocchi scaricati da uno stesso client appartengono alla stessa parte

TRASFERIMENTO DI FILES TRA CLIENTS

Selezione delle parti da scaricare

- Distribuire le richieste di parti tra le sorgenti conosciute
- Scaricare per prime le parti più rare
- Prima la prima e l'ultima parte

Part rating: viene attribuito un punteggio diverso ad ogni parte che dipende dalla

- Disponibilta' delle parti
- Parti importanti (prima e ultima parte)

INTELLIGENT CORRUPTION HANDLING

- Ad ogni parte in cui è stato decomposto il file viene associato un valore mediante una funzione hash
- L'insieme degli hash calcolati determina un hashset
- L'hash del file è ottenuto combinando questi valori
- Il client scarica l'hashset del file prima di una qualsiasi sua parte
- Ogni volta che un client scarica una parte del file, calcola l'hash H di quella parte e confronta H con il valore corrispondente contenuto nell'hash set
- Se i valori coincidono, la parte è integra e può essere messa in condivisione
- Se i valori non coincidono, si ripetere il download della parte,un chunk alla volta
- Per ogni chunk scaricato viene ricalcolato l'hash su tutta la parte, se il valore calcolato coincide con quello nell'hash set, la parte viene considerata 'riparata' e si evita di trasferire il resto dei chunks
- Questo consente di evitare di ripetere in media il 50% del download dei chunks del file

ADVANCED INTELLIGENT CORRUPTION HANDLING

- Problema dell'ICH: quando l'errore è localizzato alla fine di una parte del file, ad esempio a 8,9 MB, l'ICH riscarica tutti i chunks delle parte
- AICH= Advanced Intelligent Corruption Handling:sistema introdotto dalla versione 0.44
- AICH Utilizza un set di hash "costruiti" a partire da blocchi di 180KB (chunk) e raggruppati in una struttura "ad albero".
- Il programma crea questo albero di hash per tutti i file condivisi e lo memorizza cartella emule\config.
- Se eMule scarica un file e rileva una parte corrotta, chiede un "pacchetto di recupero" da un client che sia in possesso di un set hash A.I.C.H. completo di quel file
- Usando gli hash contenuti in esso controlla tutti i blocchi da 180KB e identifica quelli corrotti, chiedendo lo scaricamento solo di essi.

ADVANCED INTELLIGENT CORRUPTION HANDLING

- Se in un chunk un solo byte e' corrotto, eMule conserva tutti i blocchi da 180KB non corrotti, ad eccezione dell'unico blocco da 180KB che contiene il byte corrotto.
- Ad esempio, nel caso in cui il chunk corrotto si trovi in fondo al file, viene riscaricato solo l'ultimo chunk da 9.1 a 9.28 MB invece di tutti e 9.2 MB.
- Vantaggio: tramite l'A.I.C.H. è possibile recuperare parti di file senza dover essere costretti a riscaricare le parte non corrotte.
- Se se non è possibile recuperare un file tramite AICH si utilizza ICH

COMUNICAZIONE UDP CLIENT/CLIENT

- Un client *C* invia periodicamente pacchetti UDP ai clients da cui sta tentando di scaricare i files
- Il pacchetto UDP viene inviato per conoscere lo stato della richiesta di download effettuata da *C* presso *C1*
- Possibili risposte
  - La posizione di *C* nella coda di upload di *C1*
  - La coda di upload di *C1* è piena
  - *C1* non possiede il file richiesto da *C*