

Sommario



- Principi di p2p
- Esigenze dell'utente
- Trusting ed Affidabilità
- IN-FiT
 - I B-Tree
 - Protocollo ed Arrangiamento della rete
 - Trasmissione dei files
 - Un pò di conti...

- Domande e Risposte



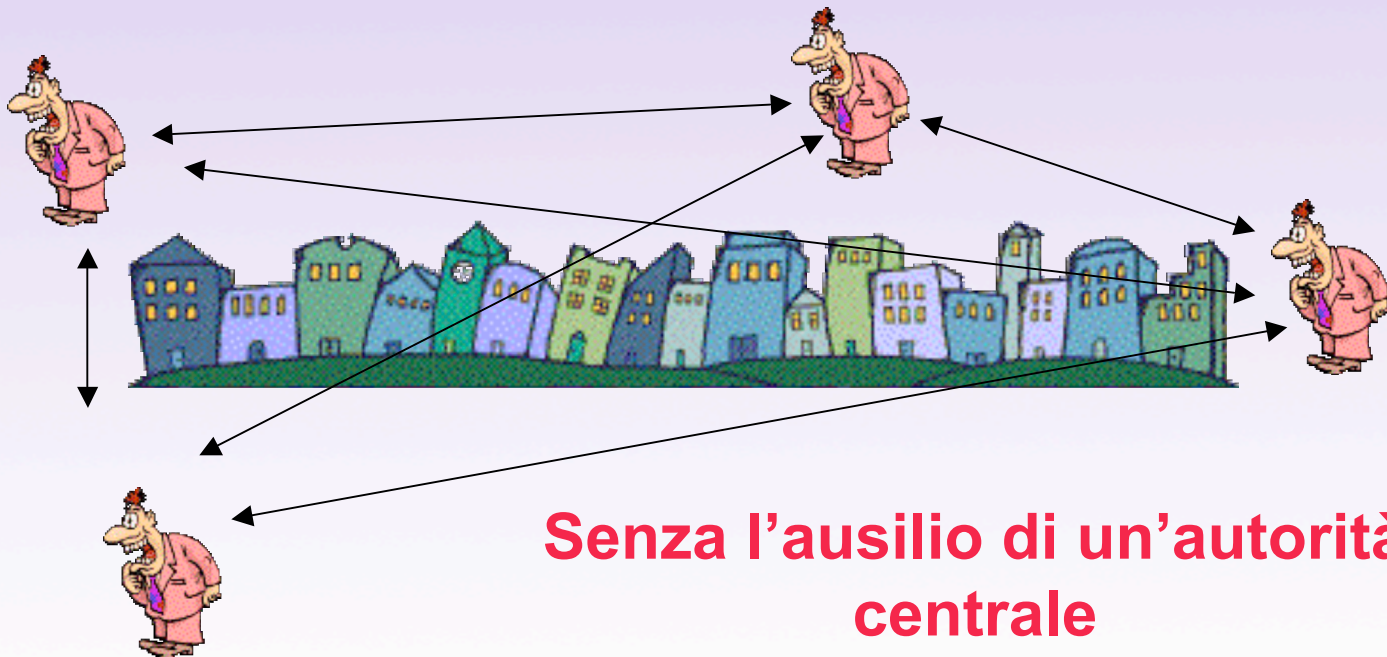


IN-FIT
IDA BASED NETWORK
FILES TRANSFER

Contesto Attuale

- I sistemi *p2p* stanno crescendo rapidamente rispecchiando le esigenze degli utenti.
- Scambiare musica, files, immagini o semplicemente effettuare conversazioni...

l'utente ha bisogno di connettersi rispettando delle basi uno-ad-uno



Le reti p2p – Cosa sono?

- Le reti p2p non rispecchiano un **singolo** protocollo di comunicazione
- Non sono semplicemente una **singola** struttura di rete
- Il modello p2p trova applicazioni in una vasta varietà di applicazioni:
(*file sharing, instant messaging, allocazione di risorse, ...*)



- Esistono punti in comune tra le varie applicazioni e protocolli
- ma molte differenze nelle specifiche implementazioni!
- Il risultato è un **groviglio inestricabile** di diversi standards
- spesso poco efficienti che mirano a risolvere **un** problema e non **il** problema



L'effetto Unidirezionale

p2p significa semplicemente: “*comunicazione eguale con eguali*”

abbastanza “**semplice**” !, *perchè tutti questi problemi?*

“*effetto di divisione dei flussi e frammentazione unidirezionale*”

- **Nodi temporanei, IP dinamici:** questi nodi possono raggiungere la rete ma, tipicamente, non sono raggiungibili da altri nodi.
- **Nodi LAN / WAN, NAT:** i nodi su una intranet dietro NAT sono generalmente non raggiungibili direttamente dall'esterno.
- **Firewalls:** i nodi dietro un firewall sono, tipicamente, non contattabili dall'esterno.

...anche se i nodi possono vedere ed usare internet attraverso i normali protocolli
il flusso dei dati è principalmente unidirezionale





L'ampiezza di Banda

ogni discussione sui sistemi p2p, presto o tardi finisce con il parlare di

allocazione e consumo di banda

esiste una legge empirica, conosciuta come legge di Gilder, che recita:

“l'ampiezza di banda, cresce circa tre volte più velocemente della potenza di calcolo”

Problema:

- **più** un'informazione è popolare, **meno** essa diventa reperibile!
- viene **saturata** la capacità di **banda disponibile** per condividere il documento.



Sicurezza ed Affidabilità



Per quanto concerne la **sicurezza** ed **affidabilità** dei dati?

- se un documento è spiacevole a qualcuno è facile eliminarlo da internet!
(viene individuata la fonte e ne viene bloccata la diffusione)
- è possibile risalire al possessore del documento in maniera facile e trasparente

Ricapitolando i sistemi p2p presentano alcuni problemi:

- **frammentazione eccessiva** dei nodi partecipanti
- **poca affidabilità** sulla persistenza dei dati in rete
- **gestione non intelligente** della banda disponibile





Il paradigma Umano

Le conversazioni per loro natura, sono **DINAMICHE!**

Quando pensiamo ad una rete di computers pensiamo ad un set di connessioni:

- **Permanenti**
- **Persistenti**

D'altra parte, le conversazioni umane sono:

- **Dinamiche**
- **Transienti**
- **Cangianti**

Caratteristiche che **devono continuare** a sussistere nei sistemi p2p.

- La topologia cambia continuamente
 - *per ogni nodo che viene, un'altro v  via e viceversa.*



Topologie di Rete



- I sistemi p2p devono essere quindi:
 - **Dinamici**
 - **Scalabili**
- Laddove le prime due features sono state implementate adeguatamente:
 - **Persistenti**
- Ogni nodo, quindi, dovrebbe:
 - **Vivere** in un determinato istante temporale
 - **Possedere** una selezione casuale di nodi vicini cooperanti con lui
- **La topologia può essere lenta o rapida nel cambiare**
 - *(in base a fattori e caratteristiche locali della rete/subnet)*





IP Dinamici e Connessioni Erranti

- La rivoluzione dei PC, seguita dal boom della connettività casalinga ha avuto immediate conseguenze nei sistemi di connessione verso internet.

La connettività statica è limitata.

- I providers possono attingere ad un **numero limitato** di indirizzi IP.
- Tipicamente si fa uso di sistemi di riallocazione dinamica per un periodo di tempo limitato degli stessi indirizzi IP (*IP sharing, DHCP, etc.*)
- Qualunque implementazione p2p che si pone come obiettivo finale l'utente, deve tenere conto di questo problema!
 - **IP Dinamico**
 - **Roaming di Connettività**

Sono problemi che DEVONO essere risolti!





Modello Client-Server

Il risultato della crescita esponenziale di internet e dei servizi ad essa legati è la progressiva affermazione di un sistema sul modello:

Client - Server

- **Web**
- **E-mail**
- **Data Storage**

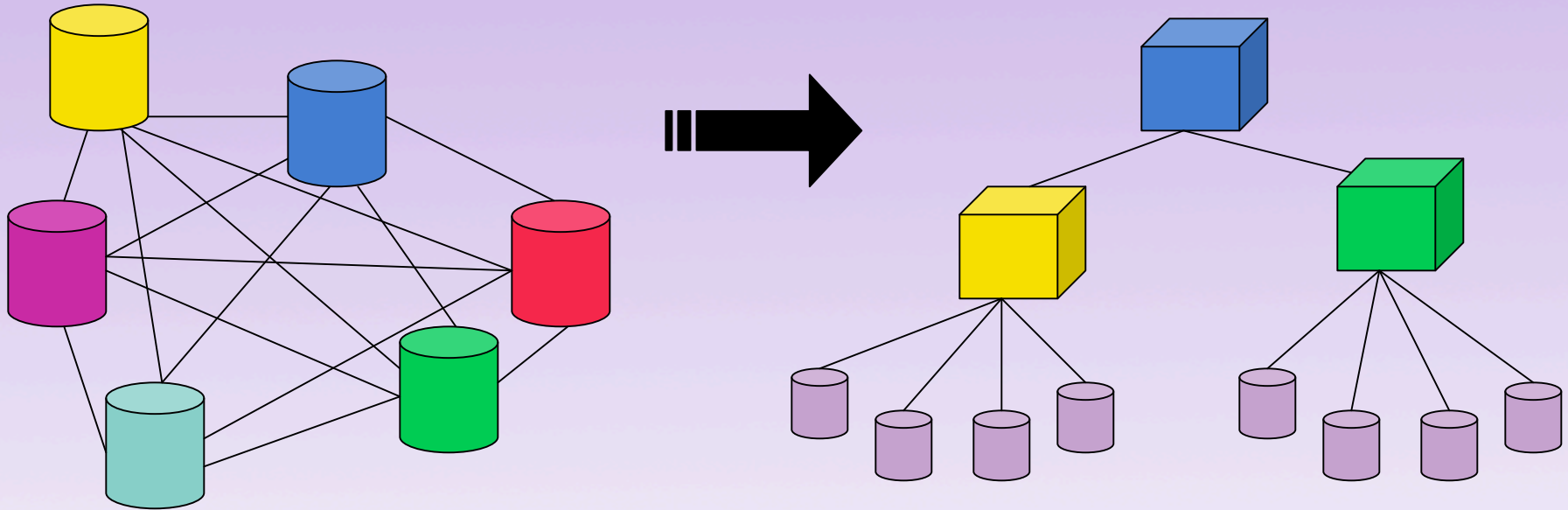
- Sono tutti paradigmi della stessa filosofia:
 - *L'utente accede a contenuti messi a disposizione e supervisionati da altri.*

Gli utenti sono visti come un contenitore passivo per i dati

- I moderni browsers web:
 - effettuano, a scopi di efficienza, una **cache** dei contenuti
 - **riducono** e **limitano** il potere interattivo dell'utente



Pregi e Difetti del CS-Model



La centralizzazione introduce **vulnerabilità** e **rischi**

- E' relativamente facile **alterare** o **distruggere** informazioni che risiedono in una locazione statica ben definita.
- E' facile **tracciare** abitudini, luoghi visitati e contenuti visionati da un utente, se tutto risiede su strutture statiche.



p2p – Power To People

- Il fatto che dati e risorse sono controllati dall'utente è una caratteristica ed una motivazione forte dei sistemi p2p odierni.

Bisogna tenere presente, comunque, che:

Condividere risorse non necessariamente implica p2p

- Il fattore discriminante, al contrario, deve essere:
 - **Come comunicano** risorse e nodi.
 - **Chi possiede** e **controlla** nodi e dati.

Se questi due fattori sono lasciati in mano all'utente finale si può parlare di:

sistemi p2p o non gerarchici o flat network models





Comunicazione tra Nodi

- I protocolli sono il collante che sorregge una rete definendo

“Come i nodi comunicano tra loro per rendere la rete funzionale”

- Sono definiti a più livelli
 - *E' possibile tracciare linee guida che suddividono e classificano i sistemi p2p*
- Possiamo raggrupparli in base al

tipo di comunicazione tra nodi

- **Message-Based.** L'obiettivo è inviare e ricevere pacchetti discreti con indirizzi.
 - Tempi e modalità con cui il pacchetto verrà instradato sono delegati ad un agente.
- **Connection-Based.** L'obiettivo è instaurare una connessione tra nodi tramite cui instradare i messaggi.
 - Questo contesto è più vicino all'idea di una conversazione telefonica
 - Il flusso è tipicamente un flusso raw di date tra i due nodi.





Modelli Architeturali

I sistemi p2p possono essere divisi in tre modelli principali:

- **Atomici** (*Ap2p*)
 - **Centralizzati lato Utente** (*UCp2p*)
 - **Centralizzati lato Dati** (*DCp2p*)
-
- UCp2p e DCp2p sono molto simili
 - Alcune differenze concettuali





UCp2p ed DCp2p

- I sistemi **UCp2p** (*User Centric p2p*) e **DCp2p** (*Data Centric p2p*) aggiungono

Forma di mediazione centralizzata

- Viene aggiunto un sistema di **repository**
- Il server **fornisce indicazioni** ai peers su **come contattarsi**

E' il principio ispiratore di **NAPSTER**

- Napster stà a metà tra **UCp2p** e **DCp2p**:
 - **messaggistica** (User Centric)
 - **file-sharing** (Data Centric)
- Una delle critiche fondamentali a questi sistemi è senza dubbio

La dipendenza da server centralizzati **Che fine ha fatto napigator.com ?**

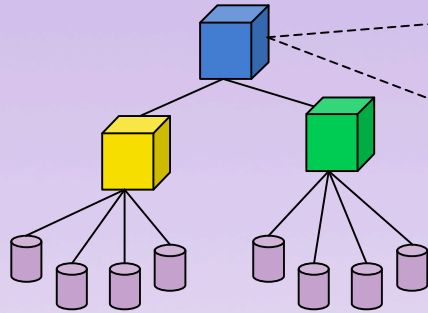
- Specifiche **vulnerabilità** e problematiche concernenti la **privacy**



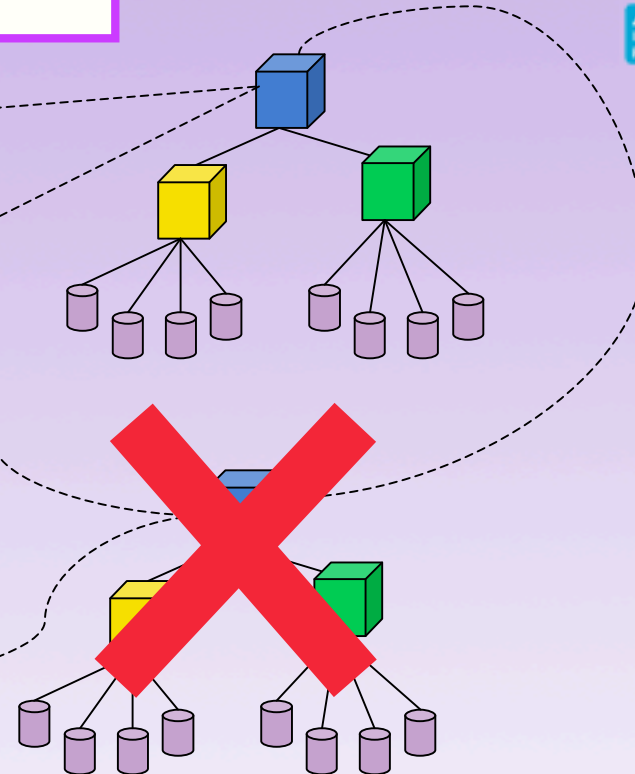
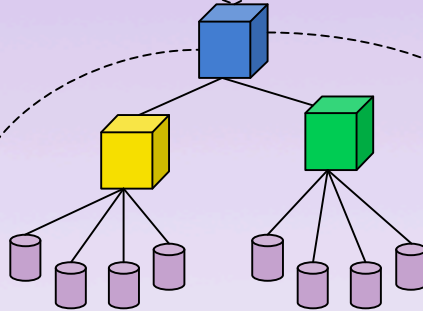


IN-FIT
IDA BASED NETWORK
FILES TRANSFER

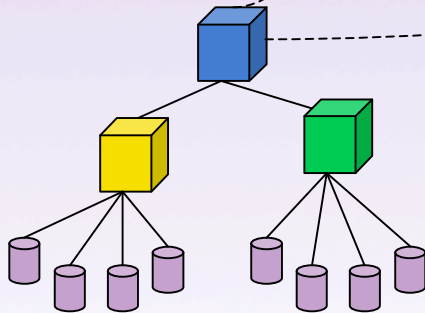
UCp2p ed DCp2p - Problemi



Un sistema di questo tipo può soffrire di gravi problematiche...



I downtime di un server significano downtime per tutti i clients ad esso connessi.



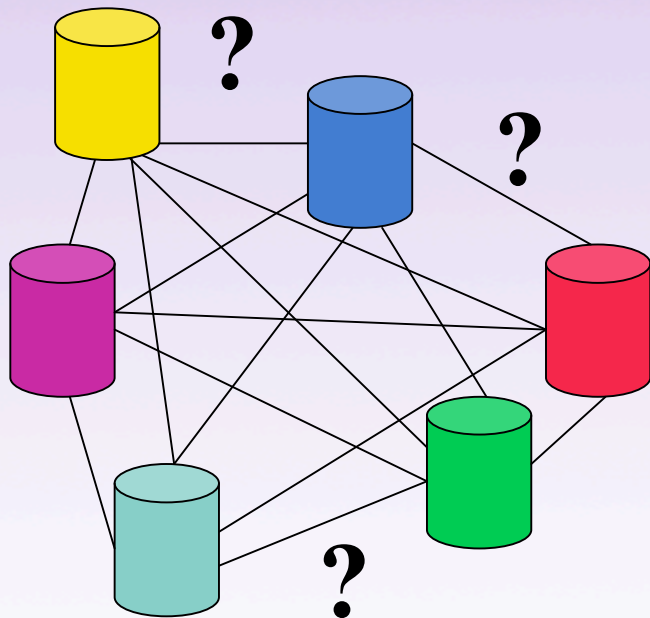
Ap2p – Il problema dell'entry point



Nel modello atomico, tutti i nodi sono in **maniera paritaria** server e client

Non esiste amministratore o arbitro di connessione

ogni nodo in questo sistema è **autonomo** e può gestire pienamente le proprie risorse.



- Questo sistema contiene al suo interno un problema:
 - **Come entrare a far parte del network?**
- I potenziali membri del network devono conoscere un modo per entrare in contatto con un altro nodo del network.
- Senza un server centrale non esiste un modo **semplice** per risolvere questo problema.

Ap2p – La sfida di Connessione



Due metodi per la risoluzione del problema della connessione alla rete:

- Utilizzare un **protocollo di broadcast** per inviare una richiesta di entrata ed attendere una risposta.
- Provare a **connettersi ad una lista** di nodi “conosciuti”, che a loro volta forniranno indicazioni circa nuovi nodi o altre possibilità di connessione.
- Con una connessione fisica all’infrastruttura del sistema (LAN)
 - La via del broadcast è, probabilmente, la più efficiente e veloce!

“Salve!, sono qui!. chi è disponibile ad ascoltarmi ?”

- Vengono create delle sessioni punto-punto tra i nodi che fanno parte del network



Ap2p e “Well Known Hosts Lists”



- I providers tipicamente **bloccano** il broadcast
- Il broadcast aggiunge un overhead **non tollerabile** per connessioni su larga scala
- La soluzione è **utilizzare una lista di nodi conosciuta** ottenuta per terze vie
 - *web, irc, email, etc.*

Il nuovo nodo può provare a connettersi utilizzando i dati della lista

- Successivamente **creerà una propria lista di nodi aggiornata**
 - In base alle richieste che riesce a monitorare una volta dentro.
- La mancanza di un server centralizzato per la connessione è un problema
- Fornisce al sistema una forma di persistenza ed alta disponibilità non alterabile

I nodi si adoperano e cooperano per mantenere viva la rete



Ap2p e “Bandwidth”



- Le maggiori critiche ai sistemi Ap2p riguardano:
 - **consumo** di banda
 - **overhead** per tenere “viva” la rete
- Legge empirica di **Amdahl**:

“la velocità complessiva del sistema è determinata dalla più lenta connessione presente nello stesso”

- **Ogni richiesta** effettuata da un nodo, viene **propagata in avanti** su ogni nodo possibile sino a raggiungere una **TTL-deadline** o una **risposta**.
- Questo tipo di considerazioni passano in secondo piano in una rete locale
 - La stessa cosa non si può dire per reti su larga scala (internet, etc.)
- I sistemi Ap2p soffrono di problemi di scalabilità dovuti ai protocolli

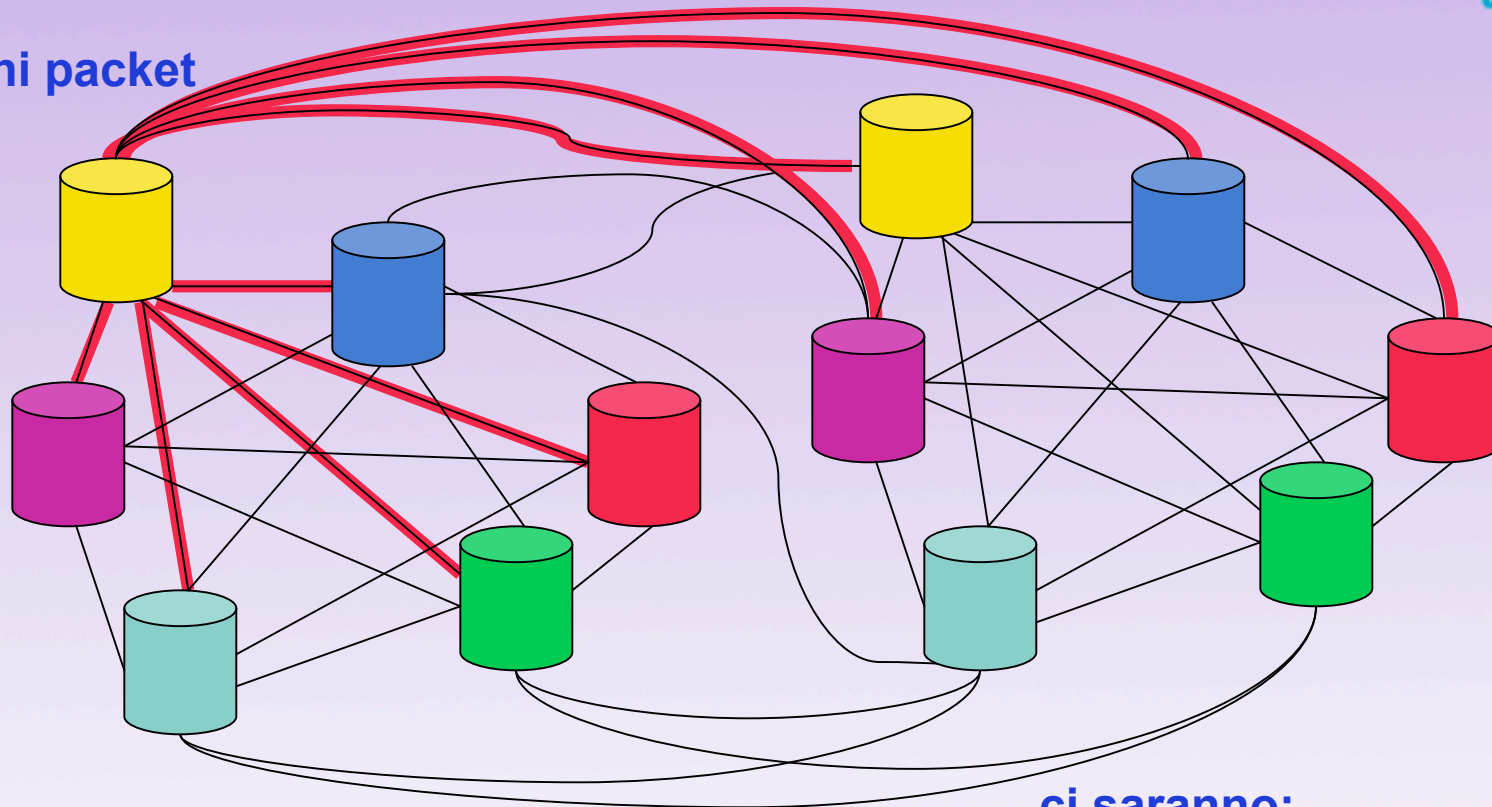
risolvono il problema in forma locale, e non in forma globale



Ap2p – Problemi Concreti



Per ogni packet



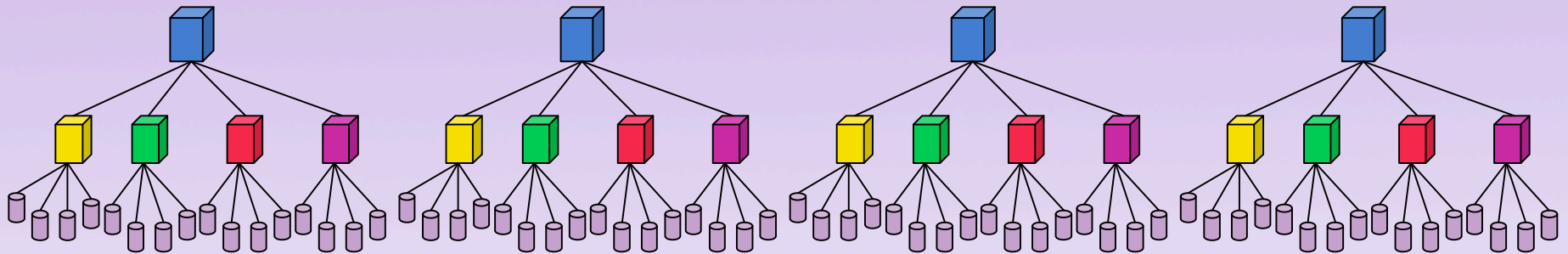
ci saranno:
 $\sim(n-1)$ packets di risposte

- **Effetto valanga**
- In una rete organizzata in maniera **caotica** il costo di mantenimento è **molto alto!**



IN-FIT
IDA BASED NETWORK
FILES TRANSFER

Il nostro approccio: IN-FiT

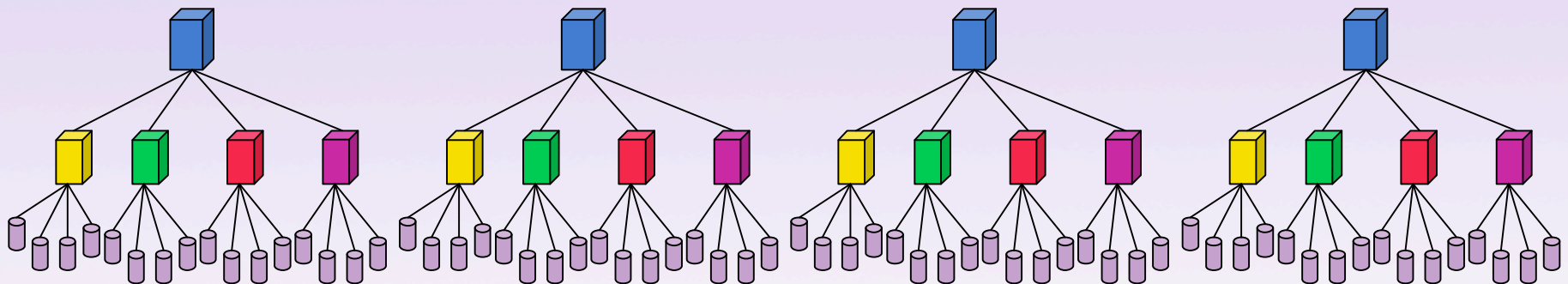


- B-Tree sono strutture *altamente scalabili*
- Il reale punto di ricerca è una *policy* che
 - *implementi* e *governi* la scalabilità della struttura.
- Un B-Tree è *altamente resistente* e tollerante ai faults hardware e/o software
- Crea la struttura, il sistema è in grado di *assicurare* la propria *sussistenza*.

IN-FiT: Scalabilità e Mantenimento



- L'albero cresce verso il basso
- Al primo livello saranno presenti N nodi
- I livelli successivi sono riempiti *progressivamente*
 - Si prova a mantenere l'albero bilanciato

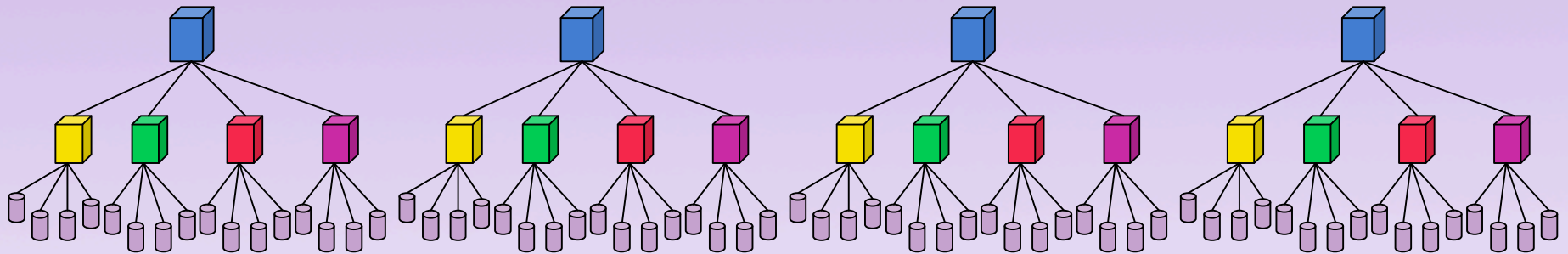


- La scalabilità è assicurata
 - L'attraversamento in verticale, impiega $O(\lg N)$



IN-FIT
IDA BASED NETWORK
FILES TRANSFER

Fault Tolerance e Confidenzialità



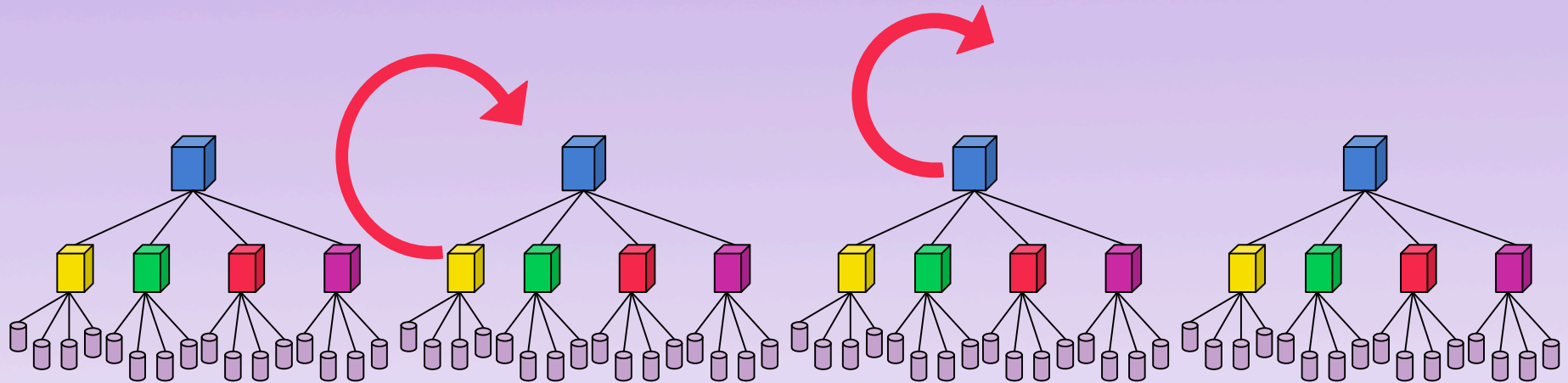
- Le operazioni sono gestiti in tempi di $O(\lg n)$
- I fallimenti sono mimizzati
 - perdere un nodo non significa perdere la rete
- Le policies devono focalizzarsi su:
 - **creazione** e **mantenimento**
 - **gestione** dei fails-over
 - rispetto della **confidenzialità**





IN-FIT
IDA BASED NETWORK
FILES TRANSFER

Reputation System



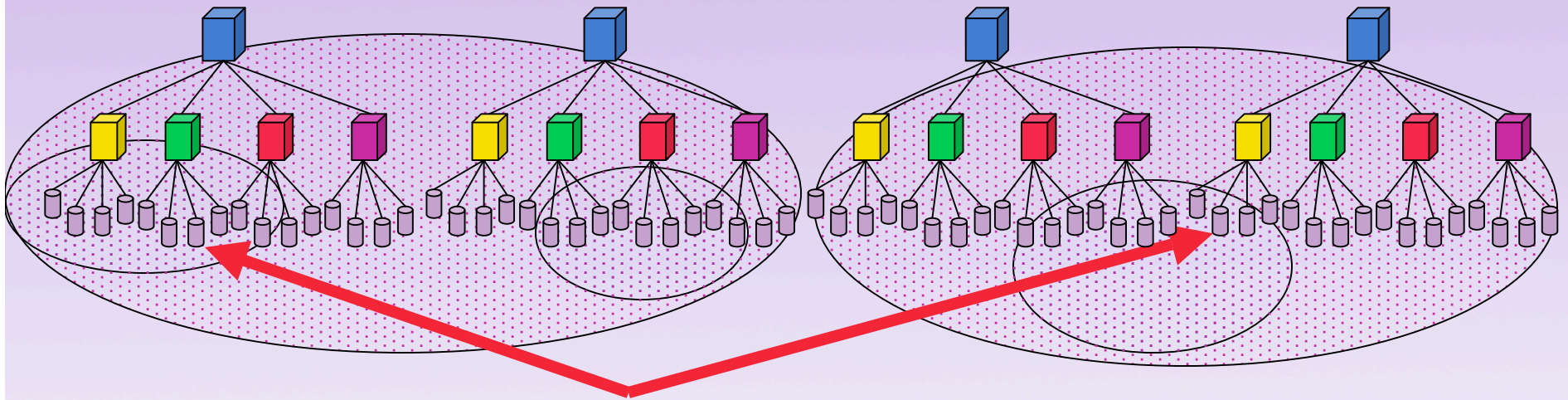
I nodi che “salgono” devono avere determinate proprietà e caratteristiche:

- **ampiezza** di banda
- **potenza** di calcolo
- **tempo** di up-time
- **CPU** disponibile
- **MPRES** abbastanza elevato



IN-FIT
IDA BASED NETWORK
FILES TRANSFER

Required Features



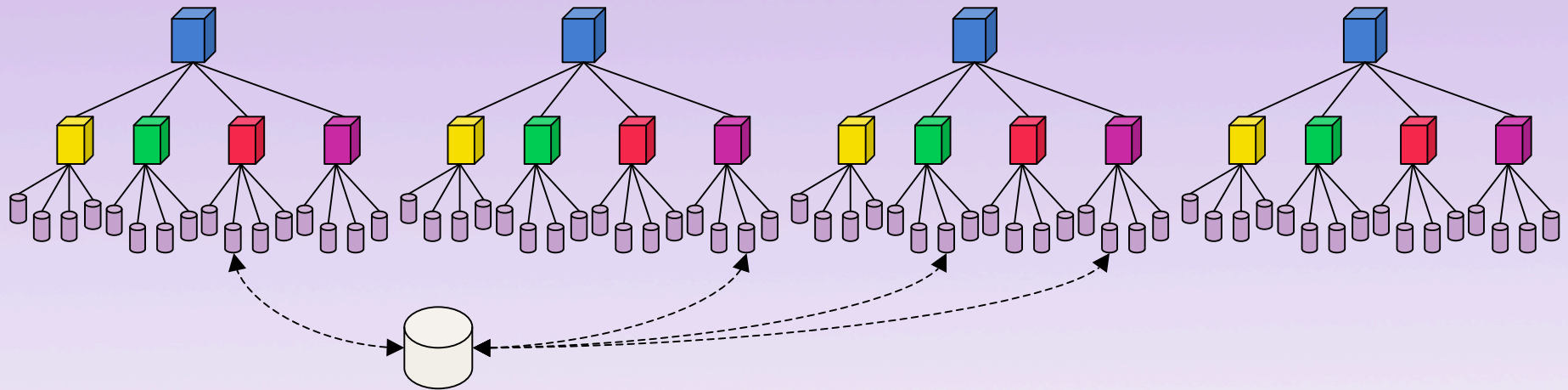
Features richieste al protocollo:

- un **collante** per le macchine che partecipano al network
- deve **possedere** funzionalità adattive per cambiamenti topologici
- deve **consentire** il reperimento di informazioni



IN-FIT
IDA BASED NETWORK
FILES TRANSFER

Come entrare nel network



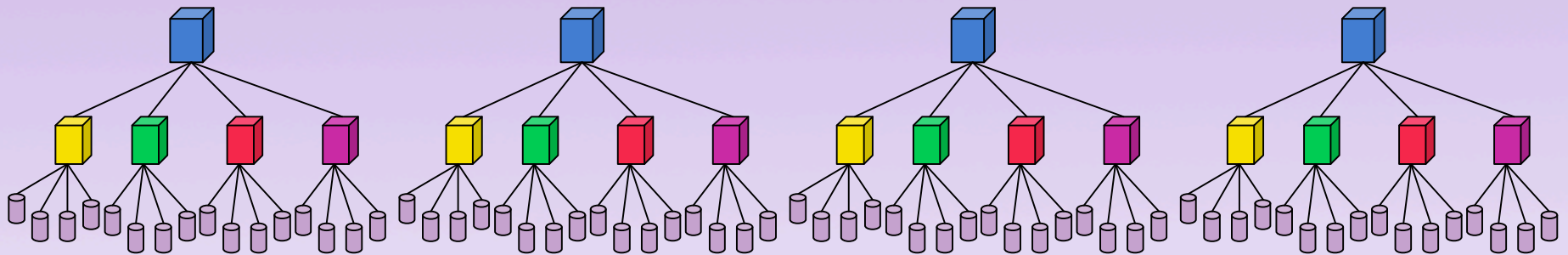
- Un nuovo nodo che vuole entrare nel network (necessita di un punto di ingresso):
 - **HELO** (*Annuncia il suo ingresso*)
- Il server risponde con una lista di nodi che mantiene in una lista dei più “**gettonati**”
 - **SRVS**<IPAddress><IPAddress>
 - Gli indirizzi sono mantenuti, nell’evenienza che nostro padre muoia
 - Durante la permanenza nel network, ampliarremo la lista.





IN-FIT
IDA BASED NETWORK
FILES TRANSFER

Come entrare nel network (2)

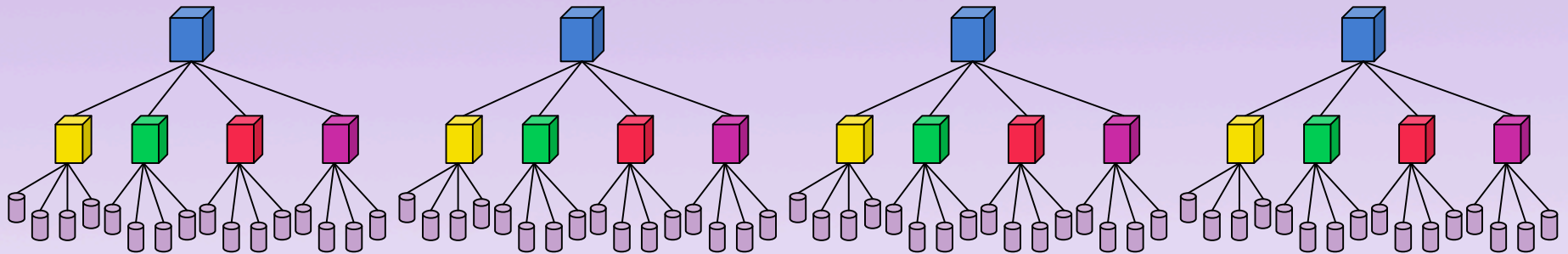


- Annunciamo la nostra lista di Chunks al padre:
 - **CHKLST**<hash>,<hash>,<hash>,...
- Nostro padre:
 - **Controlla** la sua lista
 - **Aumenta** un counter rispetto alle corrispondenze trovate
 - **Invia** la richiesta a suo padre
- Se viene superata la soglia minima, il pacchetto viene distrutto
- Altrimenti se dopo un numero **MAX** di salti, non si raggiunge la soglia
- Viene inviato indietro un messaggio di: **DEL**<hash>



IN-FIT
IDA BASED NETWORK
FILES TRANSFER

Upload di un documento

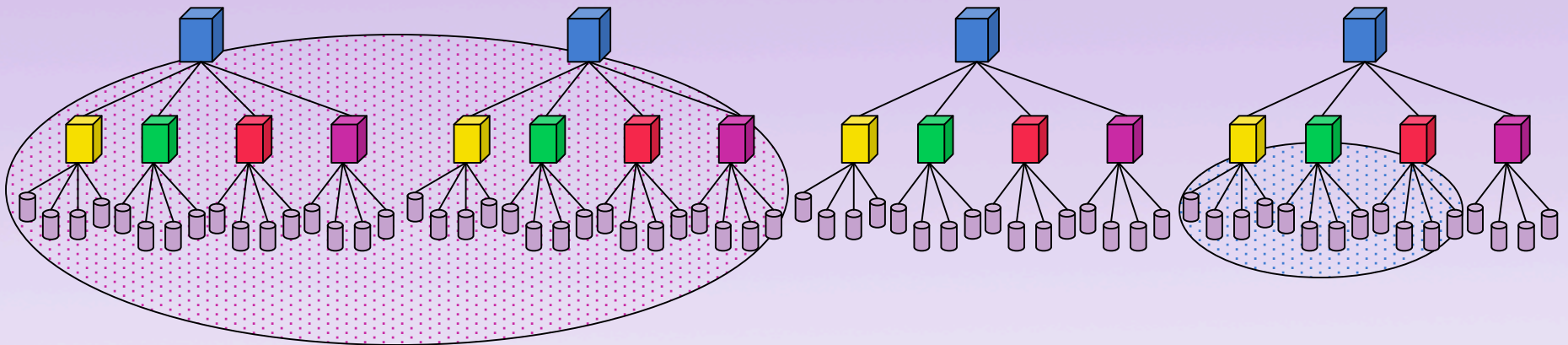


- Se vogliamo immettere un file nel network:
 - **PUTF<MIN><MAX>** (*Chiediamo una lista di nodi disponibili*)
- Il padre verifica la disponibilità dei suoi figli ed inoltra la richiesta al padre
- Ogni nodo che riceve comunicazione, contatta il mittente direttamente
- **L'anonimato** è ancora conservato
 - Ogni nodo riceve un chunk
 - Una porzione elaborata matematicamente del file originale



IN-FIT
IDA BASED NETWORK
FILES TRANSFER

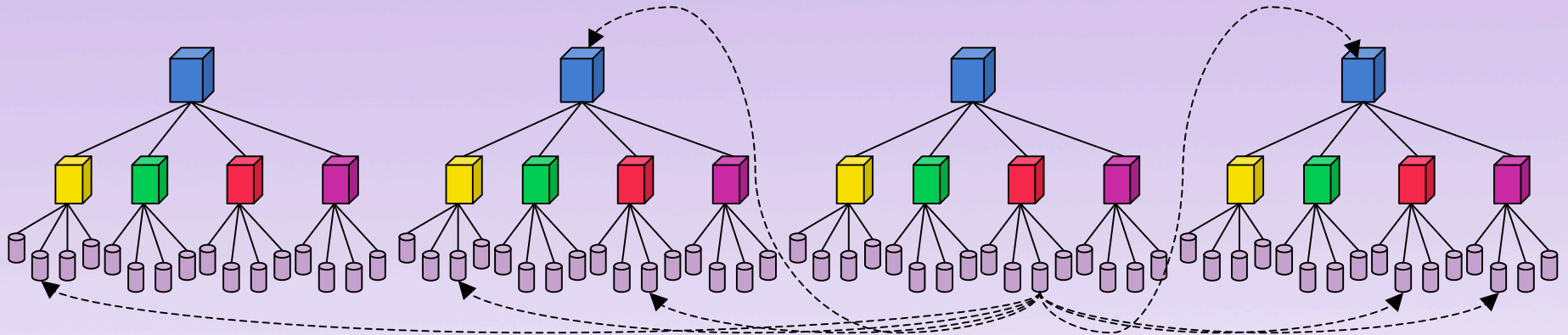
Upload di un documento (2)



- Se le trasmissioni vengono concluse correttamente
 - Il figlio annuncia l'ingresso di un nuovo documento nel network
 - **NEWF** <filename><hash1>,<hash2>,<hash3>,...
- Il padre comunica la richiesta al padre ed aggiorna la sua tabella
- Ogni zona, individuerà un certo numero di documenti



Ridondanza e Propagazione



Maggiore uso del sistema

==

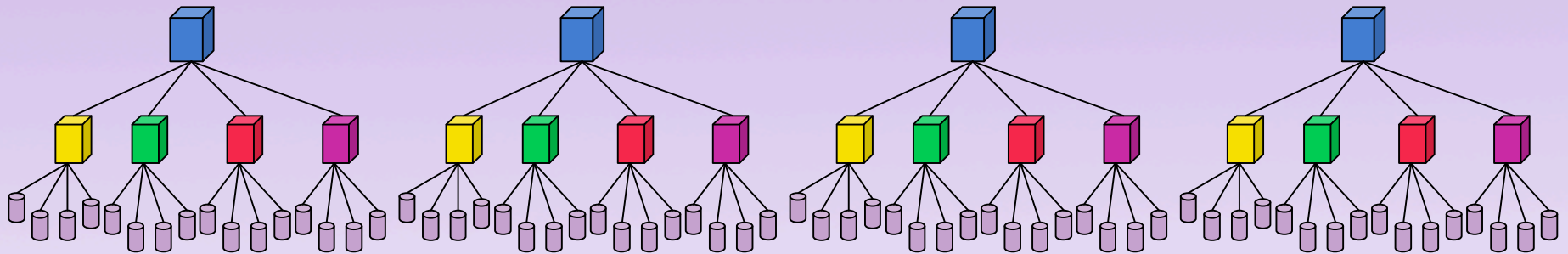
Maggiore resistenza ai faults

- Se un utente inizia il recupero di un'informazione
 - entrerà in possesso di un numero M di chunks necessari per il rebuild
 - **Aumentando** di fatto il potere di ridondanza su quel documento
- Il documento viene replicato per **“osmosi”**



IN-FIT
IDA BASED NETWORK
FILES TRANSFER

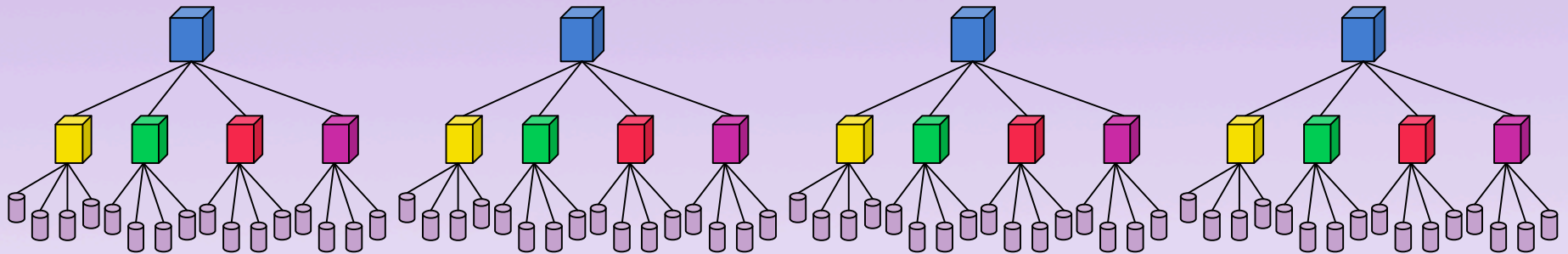
Ricerca di un Documento



- Se si vuole recuperare un informazione:
 - **SRCH**<file>
- Nostro padre
 - Controlla la propria lista
 - Incrementa il **counter di incidenza**
 - Invia la richiesta elaborata al padre



Un po' di conti...



- L'albero viene esaminato dalle foglie (nodi terminali) alle radici in tempo

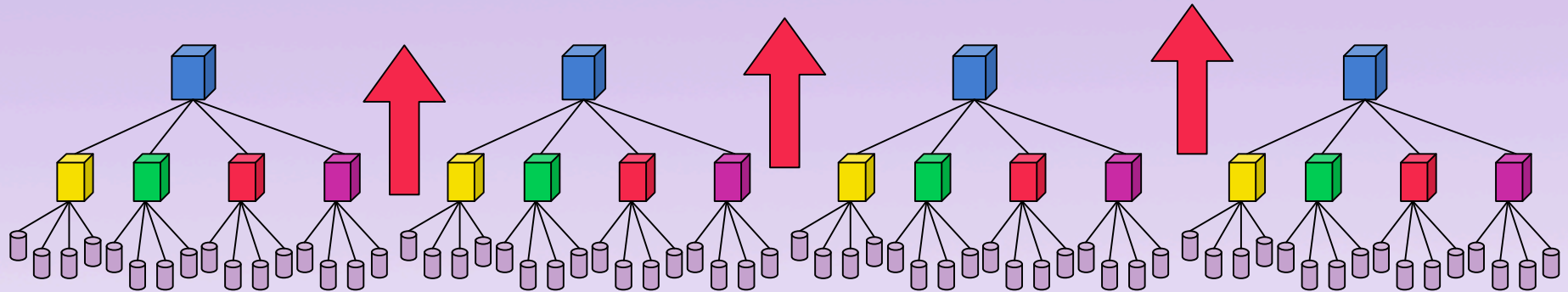
$$O(\lg N)$$

- Su di un nodo con:
 - N=5
 - 8 Livelli
- Avremo **1.953.130** nodi circa



IN-FIT
IDA BASED NETWORK
FILES TRANSFER

Un po' di conti...

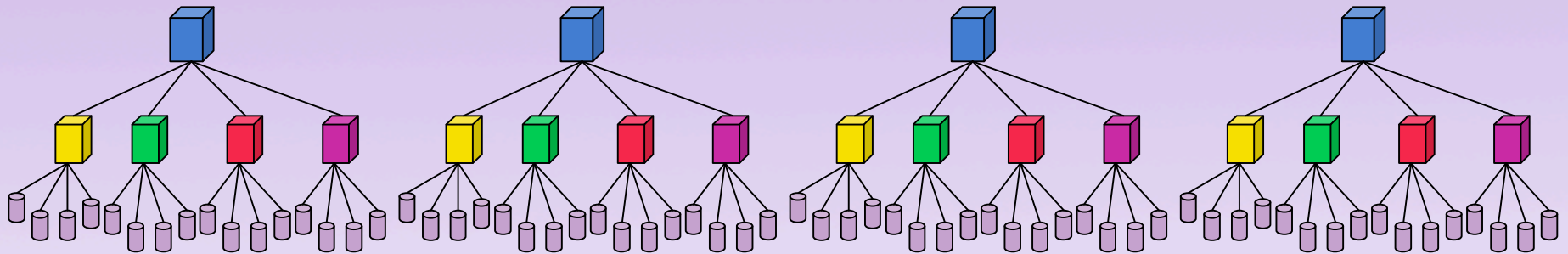


- Tutte le informazioni tendono a propagarsi nell'albero
 - Secondo una **gerarchia verticale**
 - Propagazione verso l'alto
- I nodi bassi **vengono sgravati**
 - Tipicamente sono quelli con meno risorse
- Solo I nodi con particolari caratteristiche salgono



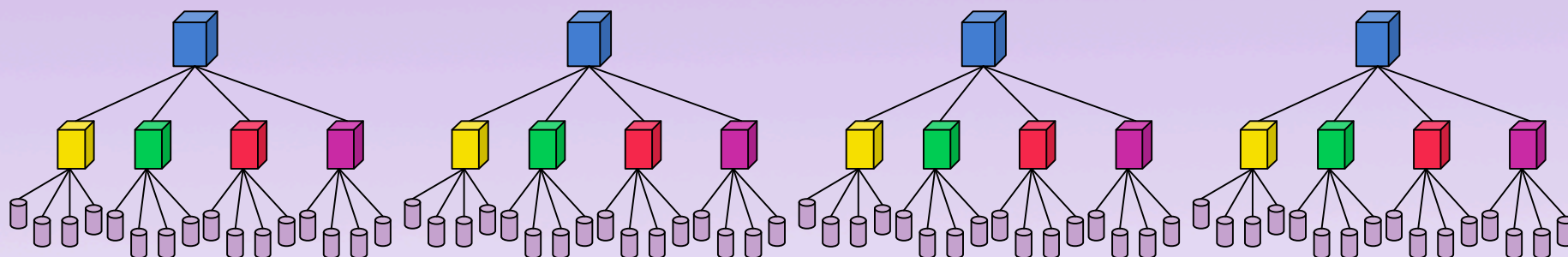
IN-FIT
IDA BASED NETWORK
FILES TRANSFER

Un po' di conti...



- Il ***fault tolerance*** è ragionevolmente garantito:
- Se un documento necessita di
 - **10** chunks minimi per essere ricostruito
 - **30** chunks sarà il valore di ridondanza
- Su una rete con **2.000.000** di nodi (**8** livelli ed **N=5**)
 - basteranno **400** download distinti
 - per superare **4.000** chunks (*valore di ridondanza*)

Anonimato ed Aspetti Legali

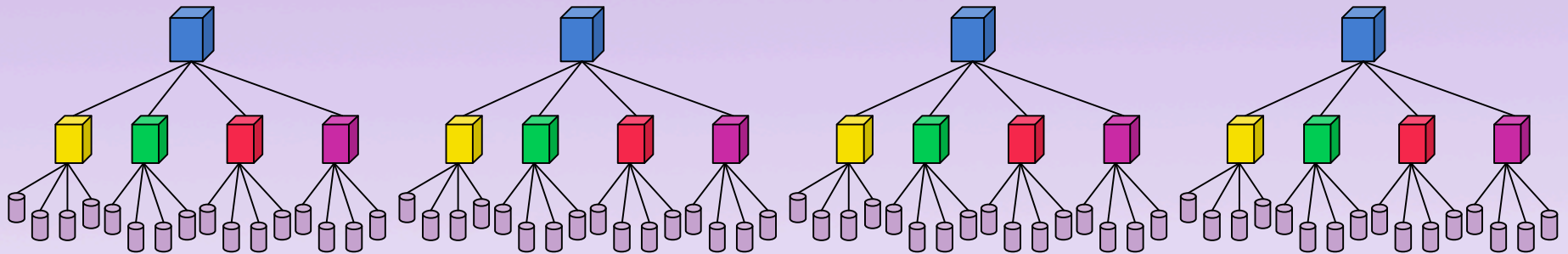


- L'anonimato è garantito:
 - Ogni nodo conserva solo chunks **non identificabili**
 - Se inviamo un chunk ad un nodo di cui conosciamo l'IP non è **GRAVE!**
 - Se riceviamo un chunk da un nodo di cui conosciamo l'IP non è **GRAVE!**
- Quel nodo non è imputabile di **nessun reato**
- **Non possiede** l'informazione se possiede un chunk
- Se non conosce l'informazione che trasporta **non potrà** alterarla
- **IDA** permette di fare **tempering** dei dati



IN-FIT
IDA BASED NETWORK
FILES TRANSFER

MPreS – MostPopularReputationSystem

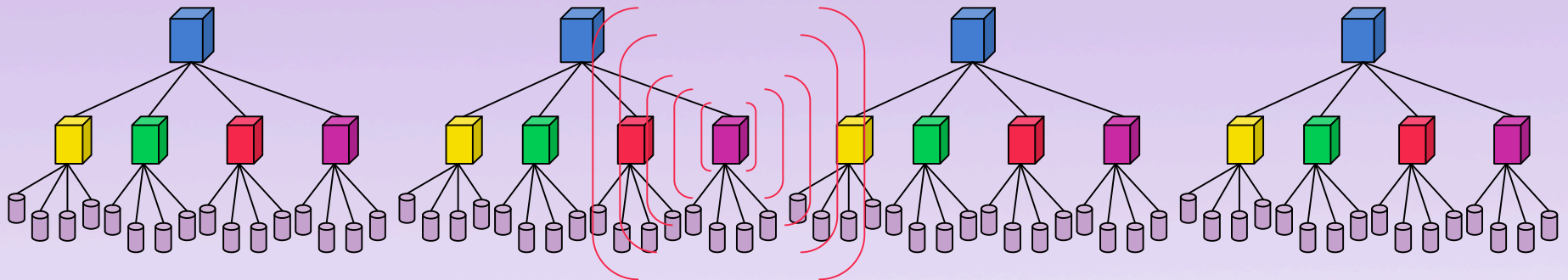


- E' necessario un approccio **non statico** e **non gerarchico** per la reputazione
 - (*ebay, payforall, etc., - Il server è garante*)
- Se un'**informazione** è **comune** al 75% dei nodi
 - possiamo concludere che l'informazione trovata è **relativamente sicura**
- **Esprimiamo** un giudizio sui nodi in base ad una classifica dei files contenuti
- Ogni documento è funzione hash del suo contenuto
 - *non è possibile "barare" sul contenuto*



IN-FIT
IDA BASED NETWORK
FILES TRANSFER

MPreS – MostPopularReputationSystem



- Ogni nodo possiede **memoria** dei nodi “**conosciuti**”
- E' possibile **cambiare opinione** su di un nodo, in seguito a **gravi illeciti**
- Ogni zona può **scambiarsi opinioni** sui nodi, creando una **reputazione di zona**
 - **NRZ** - (*Network Reputation Zone*)
- Zone diverse hanno “**opinioni**” diverse e **metri di misura** differenti
- “**Conoscere**” di persona un nodo, potrebbe **cambiare** le nostre idee.





Dove trovo IN-FiT ?

- E' un progetto di ricerca
- Sarà rilasciato con licenza:
 - BSD *oppure...*
 - GPL
- Ancora non è stato deciso!
- Il progetto è compilato per:
 - Linux (x86 / PPC) - (*console daemon / GTK+ Client*)
 - MacOS X (10.3.2+) - (*console daemon - stable*)
 - MacOS X (10.3.2+) - (*GTK+ Client - **unstable/development***)
- Ci sono ancora molti **problemi** da risolvere:
 - Design Topologico
 - Policy and Network Management
 - Bandwidth Manager
 - FileSharing Layer





References

- **freenet** - <http://freenet.sourceforge.net>
(I. Clarke, O. Sandberg, S. Miller, M. Toseland)
- **freehaven** - <http://www.freehaven.net>
(M.J. Freedman, D. Molnar)
- **gnutella** - <http://www.gnutella.org>
(Misc / Nullsoft inc.)
- **mute** - <http://mute-net.sourceforge.net>
- Costantino Pistagna <valvoline (at) s0ftpj (dot) org>
<http://www.s0ftpj.org>

