

## Documentazione progetto di Reti Informatiche

E' già stata scelta la politica di integrazione dei peers nella rete tramite un algoritmo estremamente efficiente e già testato che permette l'inserimento e la ricerca all'interno della rete nell'ordine di  $O(n)$  (very nice). Questo è descritto nel file [Prototype/soluzione per graph.txt](#)

E' necessario un sistema di IO/Multiplexing per la gestione contemporanea dei flussi di controllo da tastiera e da altri peers. Questo dovrebbe essere già presente e testato nel progetto di Antonio.

In secondo luogo, potrebbe convenire sviluppare una libreria di utility per le comunicazioni fra macchine (in quanto non mi ci vedo a scendere il girone infernale di tutta la programmazione distribuita ogni volta). Fortunatamente le slides di laboratorio del Pistolesi a riguardo sembrano piuttosto buone e potrebbero già contenere tutto quello che serve ad un livello di astrazione sufficiente.

E' ancora necessario decidere come organizzare la comunicazione tra i peer in termini di registri e richieste pendenti.

Ad ogni modo non ha **assolutamente alcun senso** lo scambio di dati tra i peers, in quanto è sicura la duplicazione di dati nel lungo termine. Al limite è comprensibile (e probabilmente richiesto) lo scambio di soluzioni a richieste di elaborazione già poste, i.e. è necessario un registro apposito per le domande già poste.

E' anche necessario un registro specifico di soli dati completi, così che anche se non è salvata in cache la risposta specifica ad una determinata domanda, sia comunque possibile calcolarla nel caso si abbiano già tutti i dati completi a disposizione (senza bisogno di contattare altri peers).

Ci sono due possibilità:

- possiamo rifiutare completamente l'esistenza di cicli
  - l'inserimento di un nodo viene fatto su distanza euclidea con le due coppie di cifre della porta del peer come coordinate cartesiane
  - questo dovrebbe evitare la necessità per ogni peer di tenere traccia delle richieste propagate ma che non hanno ancora avuto risposta. Ma credo che non sia così, che in ogni caso sia necessario salvare quest'informazione perché mi sembra brutto ipotizzare se un flood ha raggiunto tutti i nodi di una rete solo aspettando per un certo timeout senza o con alcune risposte. Allora si farebbe prima per ogni nodo di aspettare a propagare indietro l'informazione solo se ha avuto conferma di ricezione ricorsiva da tutti i suoi vicini.
  - all'uscita di un nodo si riferisce al vicino euclideanamente più prossimo tutti i propri altri vicini e agli altri, il vicino più vicino. In questo modo si mantiene l'assenza di cicli e si dovrebbe creare il collegamento più efficiente perché passa dal nodo più vicino al «baricentro».
- altrimenti continuiamo per la stessa strada, ossia con la possibilità di cicli

- l'algoritmo di inserimento è quello magico citato sopra e descritto nel file