

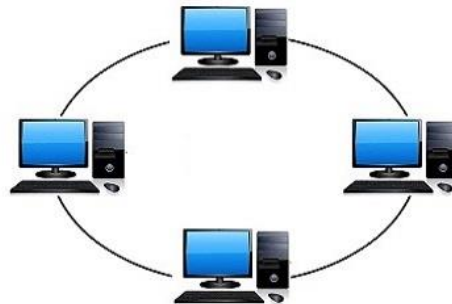
Progetto reti informatiche

Scelte progettuali:

Struttura della rete – Struttura ad anello.

Vantaggi: Semplice implementazione, minore spazio occupato nei peer per variabili di stato.

Svantaggi: Flooding oneroso che cresce linearmente insieme al numero di peers.



Collegamento alla rete – Il collegamento alla rete da parte dei peers avviene tramite lo scambio col discovery server di due messaggi UDP, uno inviato dai peers in cui comunica al DS di volersi connettere alla rete, mentre il secondo è la risposta che il DS invia al peer comunicandogli che accetta la richiesta e allega nello stesso messaggio le porte dei suoi due vicini: il primo è quello a sinistra nella rete mentre il secondo quello a destra.

Scollegamento dalla rete – Ci sono due modi in cui un peer può disconnettersi dalla rete, uno eseguendo il comando stop, tramite il quale il peer procederà a scollegarsi dopo aver inviato ai vicini i suoi record del giorno odierno. Questo serve a tutelare da eventuali mancate riconessioni, in quanto un peer che si sconnette ‘consapevolmente’ potrebbe non rientrare il giorno successivo, per evitare la duplicazione dei dati il peer svuota i suoi register. Da notare il fatto che non viene persa la possibilità di capire quale peer ha inserito quale valore grazie ai register duplicati esistenti nella sottocartella ‘saved’, che a differenza degli altri non vengono svuotati e vengono aggiornati solamente dopo una add.

L’altro modo di scollegarsi è tramite una ‘esc’ del DS, che ho interpretato come un’operazione più drastica da voler effettuare quando si sospende l’utilizzo della rete. Infatti dopo una ‘esc’ del DS i peer verranno disconnessi uno ad uno fino alla chiusura della rete. Questo è necessario perché se la rete non venisse fermata un peer potrebbe voler eseguire una ‘stop’ e senza il discovery server non c’è modo di avvertire gli altri peer di questa disconnessione o ad un eventuale riconnessione del DS non c’è modo per lui di sapere se qualche peer fa ancora parte della rete se non proibire l’uscita fino ad un suo eventuale rientro.

Connessione tra i peer – Per quanto riguarda lo scambio di messaggi tra i peer ho deciso di implementarla tramite una connessione TCP in quanto questa garantisce una maggior sicurezza rispetto alla concorrente UDP. Inoltre in particolare ho optato per una connessione TCP non persistente in quanto non è necessario mantenere delle connessioni per tutto il tempo di connessione del peer dalla rete, sarà sufficiente per il tempo necessario al calcolo dei dati aggregati o al salvataggio dei dati di un peer che invoca la ‘stop’. Inoltre essendo una rete peer-to-peer me la sono immaginata molto “dinamica” e senza una regola che fissa la durata della permanenza di un peer, questo potrebbe significare un continuo cambio dei ‘neighbor’ e creare nuove connessioni potenzialmente inutilizzate per la maggior parte del tempo sarebbe inutilmente oneroso.

Calcolo dati aggregati – Come richiesto dalle specifiche ho implementato il calcolo di due tipi di dati aggregati, il calcolo del totale dei tamponi effettuati, consultabile inserendo il comando `get T tipoDato data1 data2` (con date nel formato dd-mm-yyyy) e il calcolo della variazione, consultabile inserendo il comando `get N tipoDato data1 data2`. E' inoltre consentito inserire date con caratteri separatori delle varie sezioni differenti, l'algoritmo trasformerà la data in ciò che si aspetta di trovare (il carattere '-'). Gli acronimi CT e CN servono a indicare i dati 'Complessivi' del giorno, mentre i dati T e N indicano i dati tipo 'Base' ossia quelli non aggregati ai dati degli altri peer.

Dopo l'esecuzione di questo comando l'algoritmo si preparerà allo scambio di messaggi con gli altri peer, in particolare cercherà:

1. Chiede ai suoi 'neighbor' se possiedono il dato aggregato già calcolato.
 - a. Se lo hanno lo restituiscono e il peer lo mostra in output e lo salva
2. Altrimenti l'algoritmo inizia a scorrere i giorni chiedendo prima ai vicini se hanno un dato parziale per il giorno già calcolato.
 - a. Se lo hanno lo mette da parte e prosegue col giorno successivo
3. Altrimenti procede ad effettuare la richiesta di flooding per calcolare il dato parziale per il giorno di oggi che salverà per evitare nuovi flooding su questo giorno e poi ritorna al passo 2 per il giorno successivo.

Questo comportamento ci permette di limitare ciò che abbiamo visto essere lo svantaggio principale della nostra rete ossia: il tempo di flooding che cresce linearmente con il numero di peer connessi alla rete.

Questo dato parziale per il giorno odierno verrà salvato dal peer che ne ha richiesto il calcolo nel register del giorno richiesto in modo da renderlo fruibile a tutti i peer che lo vogliono.

Scambio di messaggi – Lo scambio di messaggi avviene inviando una serie di pacchetti in quantità differente in base al tipo di richiesta che intendiamo effettuare.

Ci sono tre campi che verranno sempre inviati:

- Tipo: Indica il tipo di richiesta che stiamo facendo al peer interpellato (Calcolo di dati aggregati, salvataggio dati etc.)
- Info: indica il tipo di dato inviato (numero casi e numero tamponi).
- Load: Corrisponde al carico, in genere contiene una stringa con il file in cui il peer ricevente deve ricercare il dato, ciò evita di dover ricalcolare il percorso che, in ogni caso il mittente aveva già preparato per cercare se aveva il dato già pronto.

E due campi opzionali inviati da tre tipologie di richieste

- Date: Contiene il dato da sommare o a cui sommare i propri record (Usato nei messaggi di salvataggio, flooding e variazione)
- Porta: Contiene la porta del peer richiedente, serve a capire all'ultimo peer di essere arrivato in fondo al ciclo e che il dato è pronto e può essere restituito. (Usato dal flooding e salvataggio).

NOTA TEST FUNZIONALITA'

All'avvio del primo peer questo procederà a creare un piccolo insieme di dati su cui è possibile testare le funzionalità del progetto. Le porte per cui ho creato i record sono da: 5001 a 5005, mentre il DS lo avvio sulla porta 4242.

Le date per cui creo dei record sono dal 10-8-2021 al 16-8-2021 scrivendo in registri a giorni alterni (es. l'11-8-2021 non sarà presente inizialmente ma se si vuole calcolare l'aggregato sull'intervallo 10/12 agosto questo verrà creato ed inizializzato a zero, questo inoltre protegge da eventuali bug a fronte di una mancata connessione di un peer in un dato giorno).