

Laboratorio Programmazione di Rete

Laurea Triennale in Informatica

Progetto Finale - Anno Accademico 2013/2014

*GOSSIP: Un servizio per lo scambio
istantaneo/differito di messaggi su WAN*

1 Scopo del Progetto

Il progetto richiede la realizzazione di *GOSSIP*, un servizio di supporto per lo scambio di messaggi tra utenti collegati ad una rete *WAN*. Prima di usufruire del servizio, ogni utente deve effettuare *una registrazione* in cui fornisce al sistema il suo *nickname*. Il sistema verifica che il nickname non sia stato già scelto da altri utenti ed, in caso affermativo, registra i dati dell'utente in un proprio database. Successivamente, un utente può usufruire del servizio connettendosi al sistema (*login*) e quindi disconnettersi dal sistema (*logout*).

Ogni utente è caratterizzato da uno *stato di presenza* sulla rete che indica, in ogni istante, se esso è *online*, cioè ha effettuato il login, oppure *offline*.

Ogni utente definisce una propria *Lista di Contatti di Ingresso*, in cui inserisce i nicknames di tutti i mittenti dai quali è disposto a ricevere messaggi ed una lista di *Lista di Contatti di Uscita* che contiene i nicknames di tutti gli utenti a cui intende inviare i messaggi: *GOSSIP* verifica l'esistenza di un utente con quei nickname. Un utente U_1 può inserire un utente U_2 nella lista dei Contatti di Uscita solo se U_2 lo ha preventivamente inserito nella propria Lista di Contatti di Ingresso.

Nel caso in cui un utente U modifichi il proprio stato, *GOSSIP* notifica tale cambiamento a tutti gli utenti interessati.

Dopo essersi connesso al sistema, un utente U_1 può inviare messaggi ad un qualsiasi altro utente U_2 contenuto nella sua *Lista dei Contatti di Uscita*. Nel caso in cui il destinatario del messaggio sia *online* il messaggio gli viene recapitato *in tempo reale*, altrimenti il sistema memorizza il messaggio e lo recapita al destinatario al momento della sua successiva connessione al sistema.

GOSSIP realizza quindi un servizio di *Instant Messaging* per gli utenti in linea ed un servizio simile alla *Posta Elettronica* per gli utenti fuori linea.

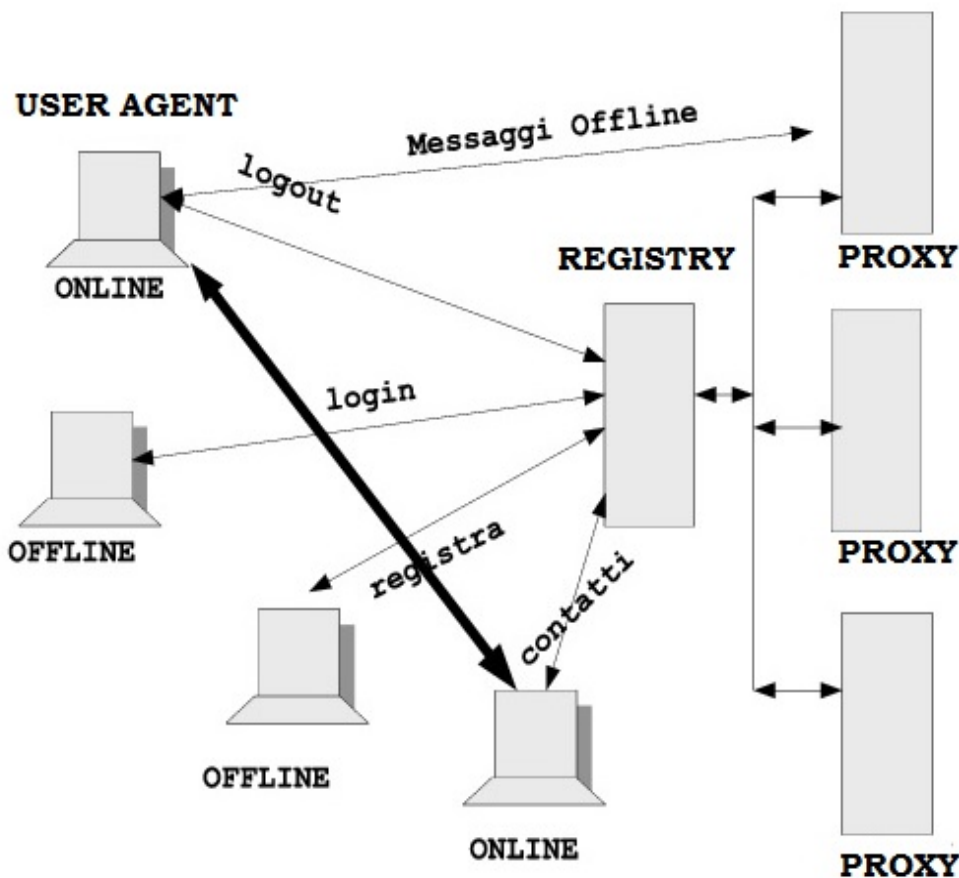


Figure 1: GOSSIP: l'architettura

2 Architettura del Sistema

Il progetto richiede la definizione di un'architettura che implementi i servizi offerti da *GOSSIP* sfruttando sia il modello client/server che quello P2P.

L'architettura comprende un *Registry*, un insieme di *Proxy* ed un insieme di *User Agent (UA)* associati agli utenti del sistema. La Fig. 1 mostra uno schema generale dell'architettura.

Il *Registry* fornisce il servizio di registrazione iniziale degli utenti, di connessione e di sconnessione dal sistema. In particolare le funzioni del *Registry* sono le seguenti:

- *Registrazione*: controllo dell'unicità del nickname fornito dal client ed eventuale registrazione
- *Gestione Liste dei Contatti*: ogni utente notifica le sue Liste dei Contatti al *Registry* che verifica l'esistenza dei nicknames dei partners e, per le

Liste dei Contatti di Uscita, la disponibilità dei partners a comunicare.

- *Connessione* : il Registry riceve le richieste di connessione degli utenti. Dopo avere verificato l'identità dell'utente U , sceglie un *Proxy* per la gestione dei messaggi offline ricevuti/inviati dall'utente. E' necessario definire una politica di assegnazione degli utenti ai *Proxy* che tenda a bilanciare il carico.

Ogni *Proxy* gestisce un insieme di utenti e svolge le seguenti funzionalità:

- quando il *Registry* gli notifica che l'utente U è online, comunica a U tutti i messaggi che gli hanno eventualmente inviato gli altri utenti nel periodo di tempo in cui U è rimasto offline
- durante il periodo di tempo in cui U rimane online, riceve da U tutti i messaggi destinati ad utenti offline e provvede alla loro memorizzazione

Lo *User Agent* interagisce con l'utente mediante una semplice interfaccia (non è richiesta la realizzazione di un'interfaccia grafica), con il *Registry*, con il proprio *Proxy* e con gli altri *User Agents*. Lo *User Agent* provvede ad inviare direttamente il messaggio al destinatario, se questo è online, oppure al suo *Proxy* in caso contrario.

3 Implementazione dal Sistema

I protocolli che devono essere implementati per realizzare la comunicazione tra le diverse componenti di GOSSIP devono essere progettati considerando le seguenti specifiche:

- il protocollo utilizzato dallo UA per effettuare la registrazione, per il login e per la gestione dei contatti di ingresso/uscita *deve essere realizzato mediante RMI*. Quando un utente U effettua il login, esso *registra inoltre una callback* presso il Registry, che viene utilizzata per avvertire U del cambio di stato (online-offline o viceversa) di un altro utente presente nelle sue liste di contatti di uscita.
- il protocollo per l'invio dei messaggi P2P tra gli UA deve essere realizzato *mediante UDP*.
- il protocollo per scaricare i messaggi registrati sul proprio proxy al momento della connessione a GOSSIP e per l'invio dei messaggi da un UA al proxy di un altro UA offline deve essere realizzato *mediante TCP*.

Inoltre si analizzi l'opportunità di strutturare ogni componente del sistema *mediante più thread di controllo*, utilizzando anche *tecniche di thread pooling*, se ritenuto opportuno.

4 Modalità di svolgimento del Progetto

Il progetto deve essere svolto singolarmente.

Il materiale consegnato deve comprendere:

- La stampa di tutto il codice dell'applicazione e di eventuale programmi utilizzati per il test delle funzionalità dell'applicazione stessa.
- Una stampa della relazione *in formato pdf* che descriva tutte le scelte effettuate. La relazione deve contenere
 - una descrizione generale dell'architettura del sistema. Motivare le scelte di progetto.
 - uno schema generale dei threads attivati da ogni componente e delle strutture dati utilizzate.
 - una descrizione delle classi definite
- il codice eseguibile con indicazioni precise sulle modalità di esecuzione.

L'organizzazione e la chiarezza della relazione influiranno sul voto finale dell'esame. L'utilizzo di metodologie di documentazione del software quali diagrammi UML (delle classi, di sequenza,...) sarà considerato positivamente ai fini della valutazione del progetto.

Relazione e codice sorgente devono essere consegnati sia in formato cartaceo, presso la portineria del Dipartimento, sia in formato elettronico, via e-mail al docente del corso. Gli eseguibili devono essere consegnati in formato elettronico.