

# Esercizio 2: Communicator

Si vuole realizzare un'applicazione Communicator che, installata sulle macchine di una rete locale, consenta agli utenti delle diverse macchine di comunicare tra loro. Il Communicator dovrà offrire due funzionalità:

1. Consentire la scoperta degli altri Communicator attivi sulla rete; la scoperta è effettuata inviando un messaggio in multicast sulla rete e ricevendo le risposte dei Communicator attivi.
2. Consentire la comunicazione con un determinato Communicator; la comunicazione consiste nello scambio di messaggi (stringhe) mediante il protocollo TCP.

Ogni Communicator dovrà quindi implementare un modulo Client ed un modulo Server che operano in parallelo:

- Il modulo Client consente all'utente di effettuare richieste per scoprire i Communicator attivi e per inviare messaggi ad altri Communicator.
- Il modulo Server gestisce la ricezione dei messaggi multicast e l'invio delle relative risposte, nonché la ricezione dei messaggi TCP e la visualizzazione del loro contenuto.

Un messaggio multicast di scoperta contiene l'indirizzo IP del Communicator mittente e la porta TCP usata dal modulo Server dello stesso mittente.

I Communicator riceventi rispondono inviando un messaggio TCP a quell'indirizzo ed a quella porta. Il messaggio di risposta contiene a sua volta l'indirizzo IP e la porta TCP del Communicator che risponde.

Per le comunicazioni in multicast si utilizzi il gruppo 230.0.0.1 e la porta 2000, mentre per le comunicazioni TCP ogni Communicator utilizzi una porta scelta dall'utente.

# Struttura del Communicator

- Il Communicator invia un messaggio in multicast, eseguendo il metodo **sendMcastDatagram**, verso il gruppo multicast 230.0.0.1 e la porta 2000.
  - Il messaggio inviato in multicast contiene il numero di porta TCP su cui il Communicator mittente è in ascolto per ricevere messaggi TCP
- I messaggi TCP vengono ricevuti e processati dal thread **SocketListener**
- I messaggi multicast vengono ricevuti dal thread **MulticastListener**, che esegue le seguenti operazioni:
  - Estrae il numero di porta TCP contenuto nel messaggio: tale porta è quella su cui il Communicator mittente è in ascolto per ricevere messaggi TCP
  - Invia su tale porta un messaggio contenente il numero di porta su cui esso (cioè, il Communicator ricevente) è in ascolto (per ricevere altri messaggi TCP): in particolare, la ricezione di tali messaggi verrà gestita dal SocketListener

# Communicator

