



UNIVERSITÀ DI BRESCIA
FACOLTÀ DI INGEGNERIA
Dipartimento di Elettronica per l'Automazione

Laboratorio di Robotica Avanzata
Advanced Robotics Laboratory

Corso di Robotica
(Prof. Riccardo Cassinis)

Controllo remoto di
SPEEDY

Elaborato di esame di: **Giuseppe Ferri**

Consegnato il: **25 novembre 2001**

Sommario

L'obiettivo che si prefigge l'elaborato è permettere di controllare il robot SPEEDY, da un terminale fisso della rete (macchina locale), sfruttando la connessione TCP/IP.

1. Introduzione

Il robot SPEEDY, disponibile in laboratorio, è abilitato solamente alla connessione seriale e quindi si è deciso di utilizzare un portatile (macchina remota), che con l'ausilio di una scheda WIRELESS fosse in grado di connettersi alla rete.

Il suddetto notebook si connette a SPEEDY utilizzando un semplice cavo che collega la porta usb0 del portatile alla porta disponibile sul robot.”

2. Il problema affrontato

Il primo passo è stato quello di leggere i manuali di SAPHIRA e ARIA messi a disposizione dall'ACTIVMEDIA, ponendo attenzione alla parte riguardante le connessioni TCP/IP, tra cui la libreria Socket.h e ArTcpConnection.h.

Poi mi sono documentato sulla comunicazione TCP/IP in generale.

Stabilire una connessione TCP/IP vuol dire aprire un canale di comunicazione (il socket) ad un indirizzo particolare (identificato dalla coppia indirizzo IP - numero di porta).

L'architettura di base appartiene al tipo CLIENT/SERVER.

In genere quando un applicativo tenta di prendere contatto con un nodo remoto via TCP/IP (il client), su quest'ultimo deve essere presente un programma che è permanentemente in ascolto (il server).

Il server ha il compito di creare il socket e di rimanere in ascolto.

Ogni volta che una connessione viene accettata, in genere, viene creata una task che ha il compito di gestire la comunicazione con il client.

Quando la comunicazione è attivata, inviare i messaggi in rete è come scrivere dei caratteri su un file. Infatti, quando viene creato il socket, in genere la funzione restituisce un descrittore (socket file descriptor).

All'interno della cartella EXAMPLES, ho trovato alcuni programmi significativi sulla gestione dei socket e dei server.

In particolare modificando i programmi "serverHelloWorld" e "clientHelloWorld" e mandandoli in esecuzione su macchine diverse, si crea una connessione TCP/IP tra un socket di tipo server e uno di tipo client, che s'invisano fra loro dei messaggi di saluto.

Questi due programmi sono pensati per esser fatti girare sotto ARIA e non per SAPHIRA.

Esiste una conflittualità tra SAPHIRA e ARIA per quanto riguarda alcune istruzioni della libreria ArSocket che, di fatto, sono quelle che consentono la connessione remota. In particolare, ad esempio, l'istruzione *connect (const char *host, int port, Type type)*, che consente di connettersi come un client ad un server, non viene accettata se nel programma inserisco la direttiva *#include "Saphira.h"*.

Questo conflittualità non è casuale, in quanto SAPHIRA prevede sì l'eventualità di una connessione remota TCP/IP, ma solo per connettere il client ad un robot o ad un simulatore che ne faccia le veci.

SAPHIRA è di fondamentale importanza per la gestione del robot e per vedere i suoi movimenti sul video e dal suo utilizzo non si può prescindere.

Modificando la funzione AREXPORT void ArNetServer::runOnce (void) chiamata nel programma netServerSimple.h e il cui corpo si può riportare dalla libreria ArNetSever.cpp, è possibile gestire un server e la comunicazione con i client connessi. I client inviano messaggi al server in cui indicano i comandi da far eseguire al robot e il server penserà a far sì che vengano eseguiti.

Non potendo però chiamare l'applicativo SAPHIRA dal programma che gira sulla macchina locale, non si riesce a raggiungere una buona soluzione per il controllo di SPEEDY in remoto.

Anche se non fosse sorto questo problema, ne sarebbe sorto un altro.

Infatti, anche se si riuscisse a connettersi in remoto da un programma client ad un programma server, dai quali posso chiamare l'applicazione SAPHIRA, mi ritroverei col problema di connettere entrambe le macchine al robot.

Dalla macchina remota non ci sarebbe alcun problema, in quanto il portatile si collega alla porta seriale del robot, ma da quella locale, da cui voglio inviare i comandi, la situazione sarebbe ancora critica.

Infatti avrei a disposizione due possibilità:

1. collegarmi direttamente via TCP/IP al robot.
2. indirizzare l'output del server sul client.

Entrambe le ipotesi sono assurde.

La prima ipotesi è assurda perché, se il robot fosse predisposto ad essere connesso via TCP/IP direttamente, il progetto non avrebbe senso, poiché basterebbe utilizzare tale opzione dalla macchina locale e non servirebbe utilizzare alcun calcolatore in remoto.

Potremmo sia connetterci dal portatile e sia da un calcolatore in postazione fissa, utilizzando direttamente l'interfaccia di SAPHIRA, oppure scrivendo un semplice programmino.

Con poche e semplici istruzioni, all'interno di un programma scritto in C++ e compilato includendo le librerie di SAPHIRA, si realizzerebbe la connessione via TCP/IP:

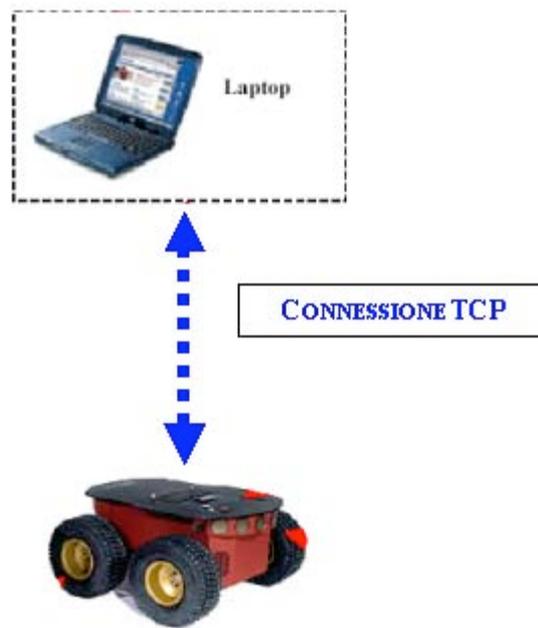


Figura 1: connessione TCP

```
ArTcpConnection tcpConn;  
ArRobot robot;  
  
tcpConn.setPort();  
if (tcpConn.openSimple())  
    {  
        robot.setDeviceConnection(&tcpConn);  
    }
```

Tutto questo però non può avvenire perché SPEEDY ha solamente la porta seriale.

La seconda ipotesi è assurda in quanto, si avrebbe una ridondanza non richiesta e non necessaria.

Non si aggiungerebbe nulla a quanto si ottiene con la soluzione finale adottata e avremmo due programmi che girano su entrambe le macchine.

La soluzione finale adottata è, infatti, concettualmente più logica e permette di non caricare la macchina locale, sulla quale non viene in realtà eseguito alcun programma.

Oltretutto all'interno dei manuali non ho trovato alcuna indicazione sull'indirizzamento dell'output sotto SAPHIRA.

Erroneamente a quanto si poteva ipotizzare all'inizio, il fatto che al menu di SAPHIRA, ci si possa collegare via TCP/IP ad un robot, non implica una predisposizione dell'applicazione a supportare un programma che esegua il ponte tra due calcolatori.

SAPHIRA è sì predisposto per connettersi via TCP/IP, ma si aspetta di connettersi ad un robot e non a un server qualsiasi e si aspetta di ricevere i pacchetti che il robot è programmato ad inviare.

Se si dovesse realizzare una connessione tra i due calcolatori con la gestione dei SOCKET, si dovrebbe:

- gestire comunicazione tra server e client,
- inviare comandi dal client al server,
- dal server far eseguire i comandi al robot,
- restituire le informazioni sulla posizione del robot al client,
- visualizzare la posizione del robot e le sue interazioni con l'ambiente circostante sul monitor del client.

Sul portatile si può sfruttare SAPHIRA per la gestione del robot, mentre sulla macchina locale da cui si opera si deve gestire tutto ex-novo, senza alcun aiuto.

Al massimo ci si potrebbe appoggiare solamente a ARIA o addirittura realizzare la connessione in JAVA.

Quindi servirebbe un data base per memorizzare i dati sul robot, costruire un ambiente grafico (GUI) per far vedere al client i movimenti del robot e sicuramente non sarebbe all'altezza di quello messo a disposizione da SAPHIRA.

Il risultato di un simile lavoro (decisamente impegnativo), però nel nostro caso, non porterebbe a risultati migliori di quanto si ottiene con l'esecuzione di X-Window in remoto.

Navigando in internet e guardando i vari progetti che sono stati realizzati in ambito connessione remota, si è scoperto che quello che si può fare è connettersi via TCP/IP, grazie alla rete, da più calcolatori, ad uno stesso robot, ma ognuno dei client si connette al robot e non ad altri calcolatori.

All'interno di una stessa rete locale, grazie a degli HUB, è possibile condividere la WIRELESS, ma i calcolatori non comunicano tra loro.

Sempre navigando, ho trovato progetti sul controllo distribuito dei robot di cui però non vengono forniti i sorgenti e le documentazioni sono descrizioni del lavoro sviluppato ben lontane dall'entrare nei dettagli. Sono progetti che hanno richiesto un lavoro complesso per gestire la comunicazione tra calcolatori e di cui molte parti sono state realizzate in Java.

3. La soluzione adottata

Dopo aver studiato le connessioni TCP/IP all'interno di SAPHIRA e aver riscontrato alcune problematiche, che spiegherò in seguito nel paragrafo 3, sono arrivato alla conclusione che la soluzione ottimale per realizzare il suddetto collegamento è connettersi dalla macchina locale della rete via TCP/IP al portatile e mandare in esecuzione remota il server X del notebook.

Lavorando con un sistema operativo LINUX, tale operazione è realizzabile.

Tale soluzione è stata anche consigliata direttamente dagli ingegneri che lavorano per l'ACTIVMEDIAROBOTS.

Si è scelto di evitare una connessione TELNET, nonostante fosse quella consigliata dalla documentazione trovata in internet, perché priva di protezione e si è optato per una connessione SSH.

Si è quindi abilitato il portatile ad essere un server SSH.

Dalla macchina locale, con poche e semplici comandi, ci si può connettere alla macchina remota e vedere l'output, come se operassimo sul terminale remoto. Per gestire il robot basterà lanciare il programma SAPHIRA e selezionare il collegamento seriale sulla porta usb0.

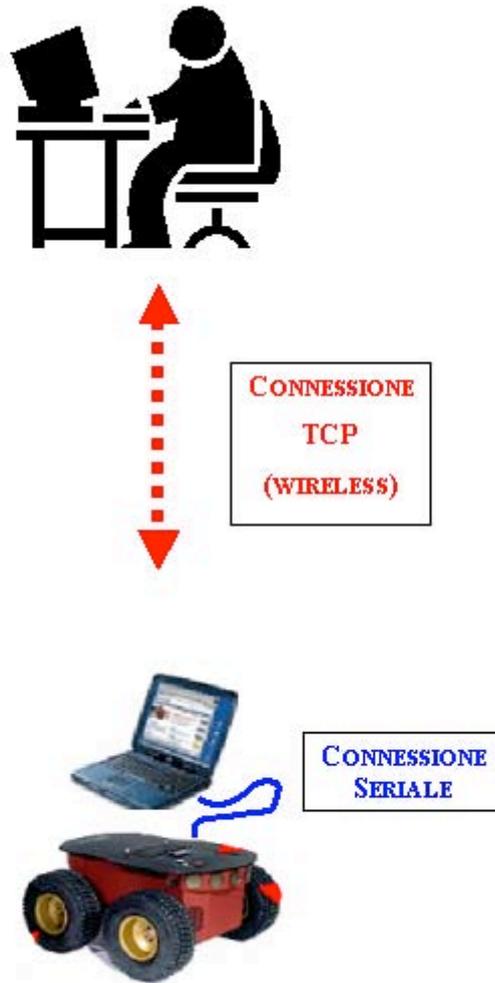


Figura 2: controllo remoto di SPEEDY

Di seguito sono elencati nella loro successione temporale, quali sono i passi da seguire per realizzare l'esecuzione in remoto di X-Window, in modo da poter pilotare da una macchina del laboratorio di Robotica, il robot SPEEDY.

Le operazioni di verifica del corretto funzionamento della scheda WIRELESS che connette il portatile alla rete e il reperimento dell'indirizzo IP del notebook si effettuano sulla macchina remota (il portatile per l'appunto).

I comandi di connessione via TCP/IP al server SSH, che gira sulla macchina remota, e quelli per l'esecuzione di X-Window in remoto, si scrivono in una finestra Xterm della macchina locale fissa, da cui vogliamo gestire il robot.

1. Inserire la scheda WIRELESS nel portatile.
2. Verificare che non ci siano cavi di rete collegati, in quanto il collegamento alla rete deve avvenire via WIRELESS, grazie alla suddetta scheda.
3. Accendere il notebook. Si dovrà attendere che il sistema verifichi che non è disponibile la connessione alla rete via cavo (eth0) e, di conseguenza, che attivi la connessione WIRELESS (eth1).
4. Verificare la connessione WIRELESS.
 - 4.1.1. Effettuare la login all'utente "remotecontrol" con password "robotsspeedy".
 - 4.1.2. Digitare il comando *ping indirizzo IP* o *ping nome di una macchina in rete*.
 - 4.1.3. Vedere se si ricevono dei pacchetti dalla rete e **PRENDER NOTA DELL'INDIRIZZO IP** che viene assegnato al notebook.
 - 4.1.4. Dopodiché uscire con il comando "Control C".

NOTA: I possessori della password di ROOT della macchina remota (portatile), possono ottenere alcune informazioni sul notebook, tra cui anche l'indirizzo IP assegnatogli. L'operazione di verifica della connessione WIRELESS, una volta entrati con i diritti privilegiati, si può fare utilizzando il comando *ifconfig*.

A questo punto, se si sono ricevuti i suddetti pacchetti, si ha la certezza che la connessione via WIRELESS alla rete è funzionante.

5. Posizionare, in maniera stabile, il notebook sul robot.
6. Realizzare il collegamento hardware tra il portatile e il robot, utilizzando l'apposito cavetto. Le estremità di tale cavo dovranno collegarsi, rispettivamente, alla porta usb0 del notebook e alla porta seriale di SPEEDY.
7. Verificare che il robot e i suoi motori per muovere le ruote siano accesi.
8. Verificare che il computer locale sia collegato al cavo di rete tradizionale.
9. Verificare che la connessione della macchina locale alla rete. Si può utilizzare lo stesso metodo visto al punto 4 per verificare la validità della connessione. RICORDARSI DI PRENDERE NOTA DELL'INDIRIZZO IP DELLA MACCHINA LOCALE FISSA.
10. Dalla macchina locale fissa aprire una finestra Xterm.
11. Tramite il comando *xhost* indirizzo IP del notebook, si comunica al computer su cui lavoriamo che il server che gira sul portatile è autorizzato ad utilizzare il nostro sistema locale. (*xhost* indirizzo IP cui si è preso nota durante l'operazione descritta al punto 4).
12. Collegarsi via rete al portatile. Tramite il comando *ssh* nome utente @ indirizzo IP del portatile, ci si connette al server come se fossimo l'utente remotecontrol.

(*ssh* remotecontrol @ IP notebook)
13. Inserire la password "robotsspeedy", come richiesto sul terminale locale.
14. A questo punto per indicare alla macchina locale che l'output del notebook va pilotato sul display locale, utilizzare il comando *export DISPLAY = indirizzo IP della macchina locale: 0*. Se, ad esempio, nel passo 9 si vede che l'indirizzo IP della macchina locale è 192.0.2.5 si dovrà digitare: *export DISPLAY=192.0.2.5:0*.
15. Se sul portatile non è stato fatto partire l'ambiente X, si può farlo dalla macchina locale con il comando *startx*.

16. Ricontrollare che il cavetto che collega la porta seriale del robot alla porta usb0 della macchina remota abbia le estremità ben inserite, perché altrimenti potrebbe essere rifiutata la connessione da SAPHIRA che comunicherebbe di non aver trovato la porta usb0.
17. Da questo momento in poi, tutto quello che facciamo sul terminale del computer locale fisso, è come se lo facessimo sulla macchina remota (il portatile appunto). Per gestire il robot la prima operazione da fare è lanciare SAPHIRA, che andrà in esecuzione in remoto sul notebook, mentre la finestra sarà visibile sul display locale.
18. Selezionare la voce Connect del menu di SAPHIRA e scegliere la connessione seriale alla porta usb0.
19. Selezionare la voce file dal menu di SAPHIRA e abilitare i motori.

A questo punto dalla macchina locale possiamo pilotare SPEEDY.

Nel momento in cui si decide di interrompere i lavori, dalla macchina locale si sconnette il robot, si esce da SAPHIRA e dalla finestra Xterm si chiude la connessione alla macchina remota con il comando *logout*.

4. Modalità operative

Questa modalità per il controllo remoto di SPEEDY è funzionale solo se si lavora sotto LINUX e se il portatile ha il server SSH attivo.

4.1. Componenti necessari

I componenti necessari sono:

- portatile con server SSH attivato,
- scheda WIRELESS da inserire nel portatile,
- cavetto che collega la porta seriale del robot alla porta usb0 del portatile.

4.2. Avvertenze

A causa di problemi non chiari di sincronizzazione (legati presumibilmente all'aspetto hardware) non sempre la connessione al robot, tramite la porta usb0, ha successo al primo tentativo.

5. Conclusioni e sviluppi futuri

In conclusione la soluzione finale dell'esecuzione di X-Window in remoto si adatta perfettamente agli scopi dell'elaborato.

Innanzitutto è una soluzione snella e semplice che raggiunge subito l'obiettivo.

Richiede ai calcolatori il minimo sforzo computazionale: SAPHIRA gira a tutti gli effetti solo sul portatile.

Purtroppo non si può realizzare uno script in cui inserire i pochi comandi che servono per connettersi dalla macchina locale al portatile, perché il server SSH richiede che la password, per l'operazione di login nel notebook, venga digitata dalla finestra Xterm sul PC fisso da cui si opera.

Indice

<u>SOMMARIO</u>	1
<u>1. INTRODUZIONE</u>	1
<u>2. IL PROBLEMA AFFRONTATO</u>	1
<u>3. LA SOLUZIONE ADOTTATA</u>	8
<u>4. MODALITÀ OPERATIVE</u>	12
<u>4.1. Componenti necessari</u>	12
<u>4.2. Avvertenze</u>	13
<u>5. CONCLUSIONI E SVILUPPI FUTURI</u>	13
<u>INDICE</u>	14