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)

Morret – Mobile Robot Remote Tunneling

Elaborato di esame di:

Alghisi Paolo, Cominardi Luca

Consegnato il: **14 giugno 2012**

Sommario

Morret è un software per la gestione remota di robot mobili in grado di operare con reti 802.11. Quando un robot mobile si collega a una rete wireless riceve un indirizzo IP privato dal DHCP server. L'indirizzo è conosciuto però solo dal server DHCP e dal robot stesso, c'è necessità di comunicare l'indirizzo IP all'esterno in modo che sia possibile raggiungere il robot. Morret si occupa proprio di questa operazione.

1. Introduzione

Nella robotica mobile vi sono una serie di problematiche con le quali i robot, e i suoi progettisti, si devono scontrare, molte delle quali sono proprie della robotica mobile e assenti nella robotica industriale. Un robot mobile, per essere in grado di muoversi in autonomia, non può essere cablato ad una postazione fissa, ha necessità di avere un collegamento radio di qualche tipo con una stazione di controllo per poter operare al meglio. Vi sono diverse metodologie per implementare un collegamento radio tra il robot e una stazione di controllo, tra di queste vi è anche la possibilità di collegare il robot a una rete wireless 802.11 e sfruttare quindi un'infrastruttura di rete già esistente.

Al giorno d'oggi le reti wireless domestiche sono molto diffuse e l'hardware per creare e connettersi a questo tipo di reti è reperibile con facilità e a basso costo. Di conseguenza si sta diffondendo sempre più l'utilizzo di questa tecnologia soprattutto per i robot mobili il cui raggio d'azione è circoscritto all'interno di edifici o nelle immediate vicinanze.

2. Il problema affrontato

Sfruttare una rete 802.11 per far connettere e gestire un robot mobile è un'ottima idea che purtroppo non è esente da problematiche. Tramite un collegamento radio punto-punto tra la stazione di controllo e il robot è possibile raggiungere il robot con facilità, purtroppo non è altrettanto semplice gestire questo tipo di problematica in reti 802.11. Quando un robot mobile si collega alla rete effettua le seguenti operazioni:

1. Si autentica all'access point con le credenziali di rete corrette
2. Invia una richiesta DHCP in broadcast e aspetta che gli venga assegnato un indirizzo IP dal server DHCP

Il punto 2 presenta una criticità da gestire in quanto l'indirizzo IP assegnato dal server DHCP al robot è un indirizzo che nella maggior parte dei casi non è noto a priori. In questo modo il robot è connesso alla rete ma è praticamente irraggiungibile poiché non si conosce il suo indirizzo IP. Se si riuscisse a rendere noto l'indirizzo IP del robot il problema sarebbe dunque risolto, nel capitolo seguente vengono illustrate diverse idee risolutive compresa quella effettivamente adottata.

3. La soluzione adottata

Ci sono due approcci principali per la risoluzione del problema, il primo richiede una configurazione del server DHCP e di un eventuale server DNS, il secondo si limita all'installazione di un software apposito sul server e sul robot.

Con il primo approccio è necessario configurare il server DHCP in modo tale che assegni sempre lo stesso indirizzo IP al robot. Questa soluzione è però poco portabile, infatti non è sempre possibile

configurare il server DHCP. Per questo motivo questa soluzione è stata scartata. Il robot infatti deve poter operare con reti 802.11 indipendentemente dall'infrastruttura di rete e dal server DHCP utilizzato.

Il secondo approccio, quello adottato, si è ispirato alla soluzione comunemente adottata per i DNS dinamici. Il robot quando si collega alla rete e riceve un indirizzo IP si occupa di comunicare questo indirizzo ad un server di controllo il cui indirizzo IP è noto. I sistemi operativi utilizzati sul server di controllo e sui robot sono tutti basati su GNU/Linux, si è deciso quindi di basare tutte le comunicazioni tra robot e server di controllo su SSH in modo da ottenere allo stesso tempo le seguenti caratteristiche:

1. Facilità di implementazione: SSH è integrato oramai in ogni distribuzione GNU/Linux;
2. Autenticazione tra i robot e il server di controllo: è possibile autenticare sia i robot che il server di controllo grazie all'utilizzo delle chiavi SSH;
3. Privatezza della comunicazione: le connessioni tramite SSH sono crittate.

Si è deciso inoltre di creare tutto il sistema con degli script bash. Lo scripting bash è risultato essere estremamente adatto su questo tipo di operazioni a livello di sistema operativo. Si è adottata una struttura client/server.

Il lato server è stato così implementato:

1. Un demone monitora una directory tramite l'utilizzo del tool "inotify". Questo tool utilizza direttamente un supporto del kernel Linux per il monitoraggio dei file. Il robot quando si collega alla rete scrive un file sul server nella directory monitorata.

   Il nome del file è l'hostname del robot mentre il contenuto del file è l'indirizzo IP del robot. In questo modo il server viene a conoscenza dell'indirizzo IP privato del robot. Quando viene notificata la scrittura di un file nella directory, il server crea un tunnel SSH direttamente con il robot e effettua una modifica del file /etc/hosts in modo tale da poter accedere al robot direttamente tramite il suo hostname.
2. Oltre al demone vi è anche uno script che effettua un'operazione di monitoraggio sulle connessioni. Quando un robot si disconnette il tunnel SSH rimane aperto sul server. Se un robot non è più raggiungibile dopo un certo quantitativo di tempo, il server se ne accorge e distrugge i tunnel SSH.

Il lato client è così implementato:

1. La connessione alla rete wireless da parte del robot avviene tramite l'applicativo wpa_supplicant. Il pacchetto wpa_supplicant contiene anche l'applicativo wpa_cli che permette di monitorare lo stato della connessione ed eseguire delle azioni specifiche nel momento in cui il robot si connette/disconnette dalla rete wireless. Quando il robot si connette all'access point viene inviata una richiesta DHCP; appena viene assegnato un indirizzo IP al robot, quest'ultimo lo comunica al server con la modalità descritta precedentemente.
2. Vi è uno script che effettua un'operazione di monitoraggio della rete. C'è la possibilità che si verifichi una condizione per cui il robot sia connesso alla rete ma il server non è in grado di vederlo. Lo stato dell'associazione alla rete wireless è relativo solamente alla connessione tra robot e access point. Se la rete retrostante l'access point è non funzionante il robot non è in grado di comunicare con il server e viceversa. Se ciò accade dopo che sia già stato creato il tunnel SSH, il server, non potendo più raggiungere il robot, distruggerà i tunnel SSH. Per questo motivo, il robot dev'essere in grado di accorgersi di questa situazione e ricomunicare al server il proprio indirizzo IP appena la rete torna in funzione.

Una volta che è stato instaurato il tunnel SSH è possibile dal server collegarsi alla porta utilizzata da Morret in localhost e tutto il traffico verrà ridiretto verso il robot. La stessa cosa avviene se ci si collega alla porta utilizzata da Morret sul robot, tutto il traffico verrà ridiretto verso il server.

4. Modalità operative

Questo paragrafo descrivere le modalità necessarie per utilizzare il software realizzato. Lo scopo è quello di permettere a un utente, anche poco esperto, di verificare in modo autonomo che quanto è stato realizzato funzioni.

4.1. Componenti necessari

Affinché Morret possa funzionare correttamente ha bisogno che siano installati sui sistemi server e client alcuni software. Oltre al nome generico del software viene fornito anche il nome del pacchetto da installare nel caso si utilizzi un sistema Debian Squeeze.

Server:

1. OpenSSH (openssh-server)
2. Inotifytools (inotify-tools)

Client:

1. OpenSSH (openssh-server)
2. Wpa_supplicant (wpa_supplicant)

4.2. Modalità di installazione

Prima di qualsiasi tipo di configurazione è necessario installare il software necessario sia sul client che sul server. I comandi da eseguire per l'installazione sono sempre riferiti a una distribuzione Debian Squeeze.

4.2.1. Server

In questa sezione sono illustrati tutti i passi necessari per una corretta installazione e configurazione della parte client di Morret.

4.2.1.1 Preparazione e configurazione

Installazione del server Openssh:

```
$> apt-get install openssh-server
```

Installazione dei tool di inotify:

```
$> apt-get install inotify-tools
```

Creazione di una coppia di chiavi SSH per il server di controllo:

```
$> ssh-keygen –t rsa
```

Copia della chiave pubblica del server sul robot in modo tale da avere accesso al robot con i privilegi di root:

```
$> scp /root/.ssh/id_rsa.pub root@robot:/root/.ssh/authorized_keys
```

A questo punto entrare nella directory server di Morret. Prima di installare il software è necessario configurare correttamente i seguenti file:

1. *src/morret.default* con l'interfaccia su cui il server è in ascolto, la directory in cui sono salvati i file temporanei dei client, le porte da usare per i tunnel. Per la directory in cui saranno salvati i file temporanei dei client è consigliato impostare una directory all'interno di */tmp* per evitare problemi con i permessi.
2. *src/rc.local* con l'interfaccia corretta (sostituire eth0, se necessario, con l'interfaccia di rete corretta)

4.2.1.2 Installazione

Per installare il programma sul server eseguire lo script *install.sh* con privilegi di root.

```
$> su -
$> chmod +x install.sh;
$> ./install.sh
```

oppure

```
$> chmod +x install.sh;
$> sudo ./install.sh
```

Durante l'installazione del programma verrano creati i seguenti file/directory:

Parametri di configurazione di Morret:

/etc/default/morret

Script di avvio di Morret:

/etc/init.d/morret

Eseguibile di Morret:

/usr/local/bin/morret

Eseguibile del monitor di Morret:

/usr/local/bin/morret_mon

4.2.1.3 Rimozione

Per disinstallare Morret eseguire lo script *clean.sh* con privilegi di root.

```
$> su -
$> chmod +x clean.sh;
$> ./clean.sh
```

oppure

```
$> chmod +x clean.sh;
$> sudo ./clean.sh
```

4.2.2. Client

In questa sezione sono illustrati tutti i passi necessari per una corretta installazione e configurazione della parte client di Morret.

4.2.2.1 Preparazione e configurazione

Installazione del server Openssh:

```
$> apt-get install openssh-server
```

Installazione di wpa_supplicant:

```
$> apt-get install openssh-server
```

Creazione di una coppia di chiavi SSH per il server di controllo:

```
$> ssh-keygen –t rsa
```

Copia della chiave pubblica del robot in modo tale da avere accesso al server di controllo con un utente non privilegiato:

```
$> scp /home/user/.ssh/id_rsa.pub root@server:/home/user/.ssh/authorized_keys
```

A questo punto entrare nella directory client di Morret. Prima di installare il software è necessario configurare correttamente i seguenti file:

1. *src/wpa_supplicant.conf* con i parametri corretti della rete wireless a cui ci si vuole collegare. Per configurare correttamente wpa_supplicant cercare degli esempi di configurazione per wpa_supplicant. (esempi e tutorial: http://hostap.epitest.fi/wpa_supplicant/)
2. *src/morret.default* con l'indirizzo ip (o hostname) del server, l'utente con cui il client si collegherà al server e il nome dell'interfaccia wireless

4.2.2.2 Installazione

Per installare il programma sul client eseguire lo script *install.sh* con privilegi di root.

```
$> su -
$> chmod +x install.sh;
$> ./install.sh
```

oppure

```
$> chmod +x install.sh;
$> sudo ./install.sh
```

Durante l'installazione del programma verranno creati i seguenti file/directory:

Configurazione di wpa_supplicant:

/etc/wpa_supplicant.conf

Directory di configurazione di Morret:

/etc/morret

Azioni da eseguire quando il client si connette alla rete wireless:

/etc/morret/function.sh

Parametri di configurazione di Morret:

/etc/default/morret

Avvio di wpa_supplicant all'avvio per la connessione alla rete wireless:

/etc/rc.local

4.2.2.3 Rimozione

Per disinstallare Morret eseguire lo script *clean.sh* con privilegi di root.

```
$> su -
$> chmod +x clean.sh;
$> ./clean.sh
```

oppure

```
$> chmod +x clean.sh;
$> sudo ./clean.sh
```

4.3. Avvertenze

Ci sono solo alcune piccole avvertenze da tenere a mente per evitare spiacevoli grattacapi. Una volta generate e copiate le chiavi ssh sul server di controllo e sul robot è necessario fare una connessione a mano dal server verso il robot e viceversa. Quando ci si collega per la prima volta ad un host, ssh chiede se si vuole accettare e ritenere affidabile la chiave ssh dell'host remoto. La connessione manuale serve appunto per accettare le chiavi.

Dal server:

$> ssh root@robot

Dal client:

$> ssh user@server

In entrambi i casi sarà presentato un dialogo simile a questo:

```
david@desk4 $ ssh admin@example.vm.bytemark.co.uk
The authenticity of host 'example.vm.bytemark.co.uk (89.16.174.65)' can't be est
ablished.
RSA key fingerprint is ce:ed:ec:11:5f:17:92:f0:5c:de:cc:ad:43:9a:18:c7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'example.vm.bytemark.co.uk' (RSA) to the list of know
n hosts.
admin@example.vm.bytemark.co.uk's password:
```

5. Conclusioni e sviluppi futuri

Con questo elaborato è stata possibile utilizzare una rete wireless 802.11 già esistente come infrastruttura di rete per gestire un robot mobile. Con un approccio di questo tipo, basato sul paradigma del DNS dinamico e tramite i tunnel SSH, risulta molto semplice creare degli applicativi di gestione del robot tramite il framework ARIA. In questo modo chi svilupperà l'applicativo non dovrà preoccuparsi della gestione della connessione alla rete ma opererà sempre in localhost. Tutte le connessioni verranno reindirizzate automaticamente tra il robot e il server di controllo. Un possibile utilizzo è quello di far effettuare tutte le operazioni di calcolo massicce direttamente al server in modo tale da aver bisogno di minore potenza di calcolo, e quindi di energia richiesta, sul robot.

Bibliografia

[1] Wpa_supplicant, home page del progetto: http://hostap.epitest.fi/wpa_supplicant/ .

[2] Inotify-tools, home page del progetto: https://github.com/rvoicilas/inotify-tools/wiki/ .

[3] OpenSSH, home page del progetto: http://www.openssh.org/ .

[4] Bash, official reference: http://www.gnu.org/software/bash/manual/bashref.html .

Sommario